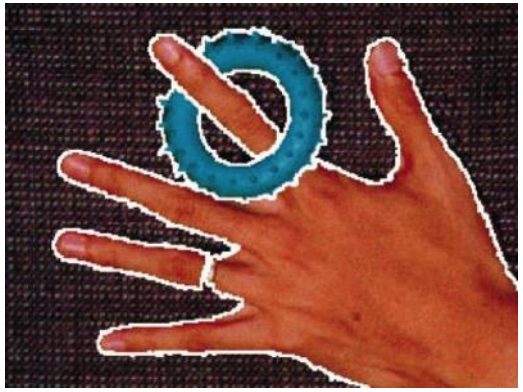
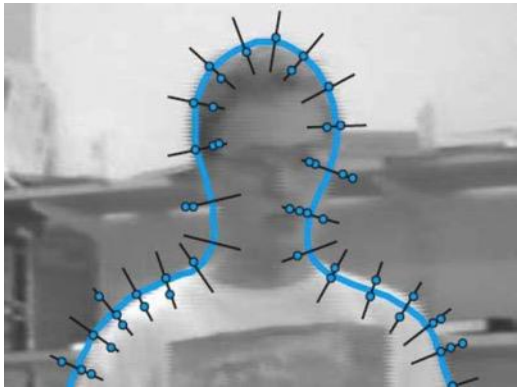




COMP 9517 Computer Vision

Segmentation

Part 1

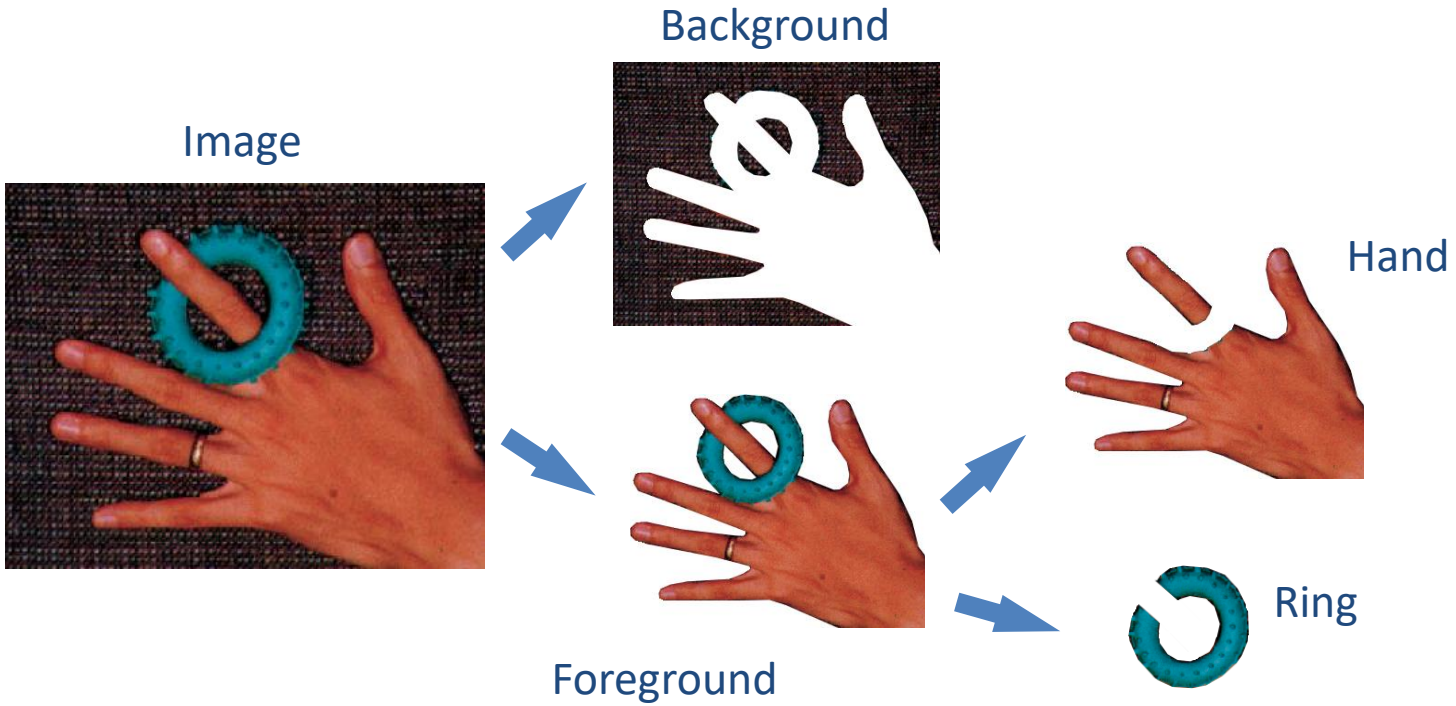
Introduction

- What do you see in this image?

Introduction

- Segmentation is the process of partitioning an image into a set of meaningful regions for further analysis

  One of the oldest and most widely studied problems in computer vision



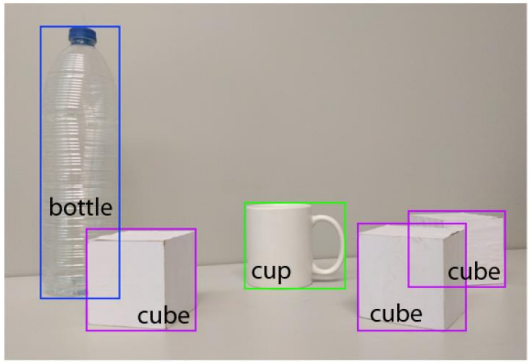
Introduction

- Region properties to facilitate image segmentation
  - Regions should be **uniform / homogeneous** in some characteristics
  - Region interiors should be **simple** and without holes or missing parts
  - Adjacent regions should have **significantly different** values in terms of the characteristics in which individually they are uniform
  - Boundaries of each region should be **smooth and spatially accurate**

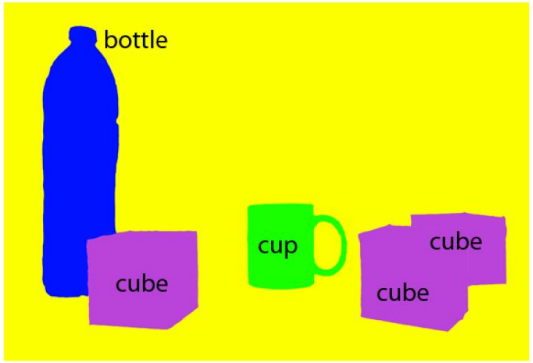
Introduction

- Different levels of region identification and segmentation

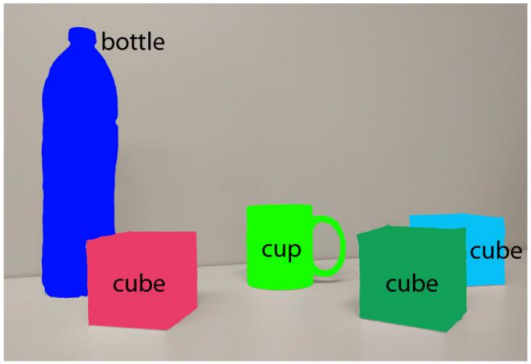
Image classification



Object localization



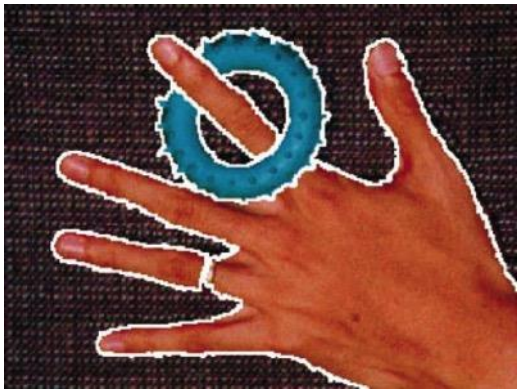
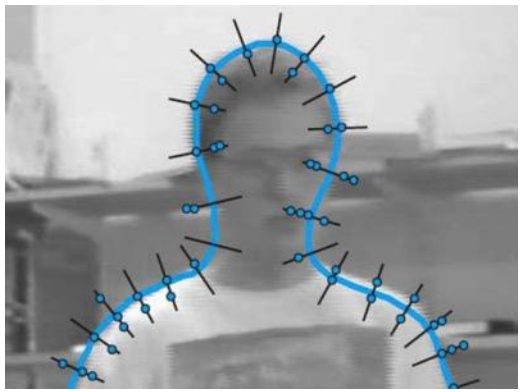
Semantic segmentation



Instance segmentation

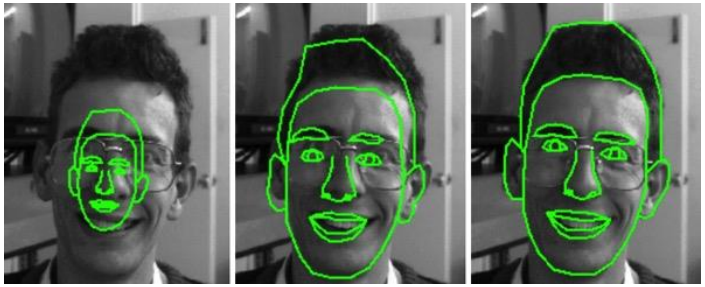
Introduction

- Segmentation approaches
  - Region based
  - Contour based
  - Template matching based
  - Splitting and merging based
  - Global optimisation based

Introduction

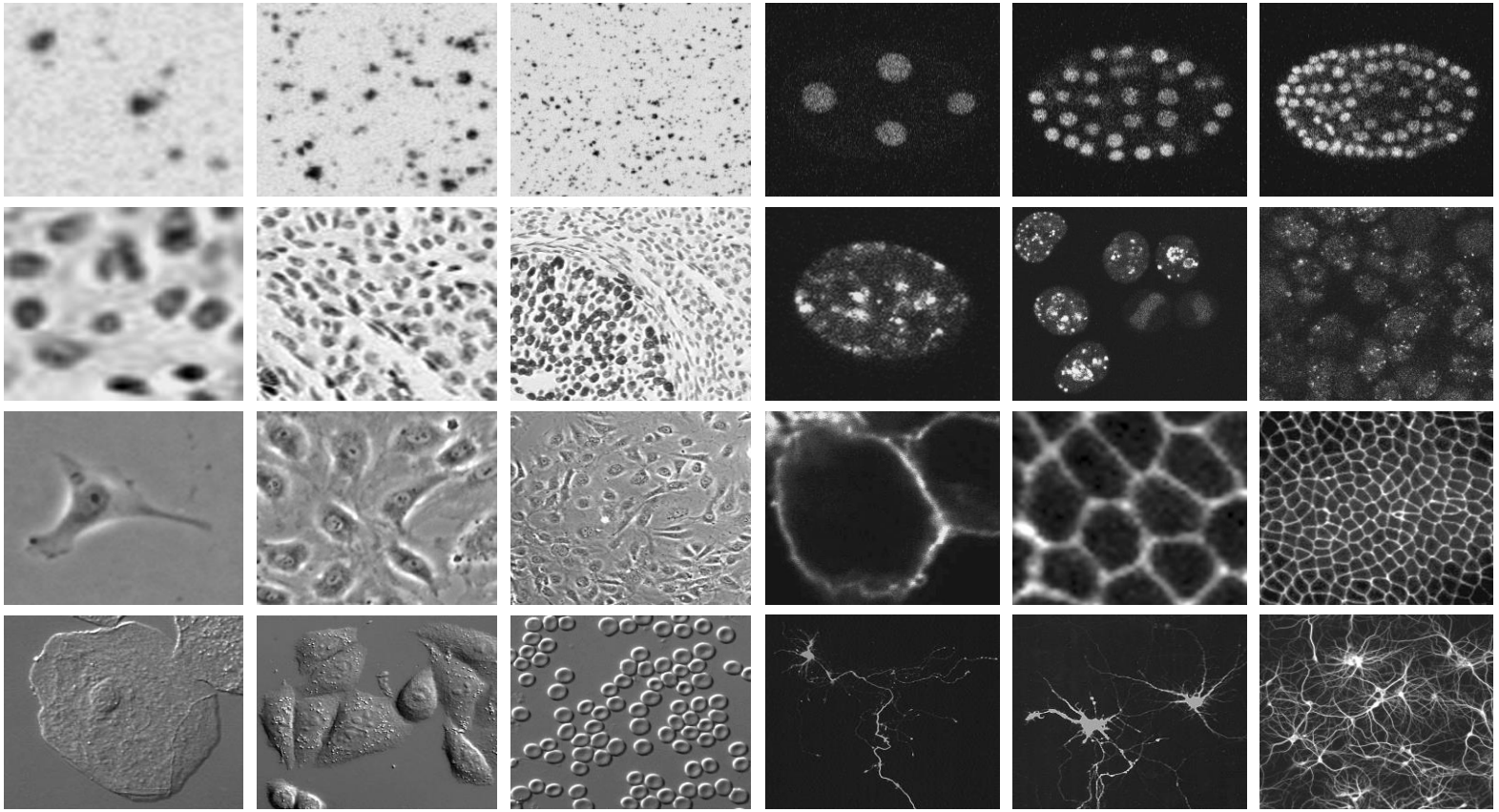
- Issues and challenges
  - So far there is **no single** segmentation method working well for all problems
  - Special **domain knowledge** of the application is typically essential for the development of successful computer vision methods for segmentation



Source: http://www.isbe.man.ac.uk/~bim/Models/asms.html

Introduction

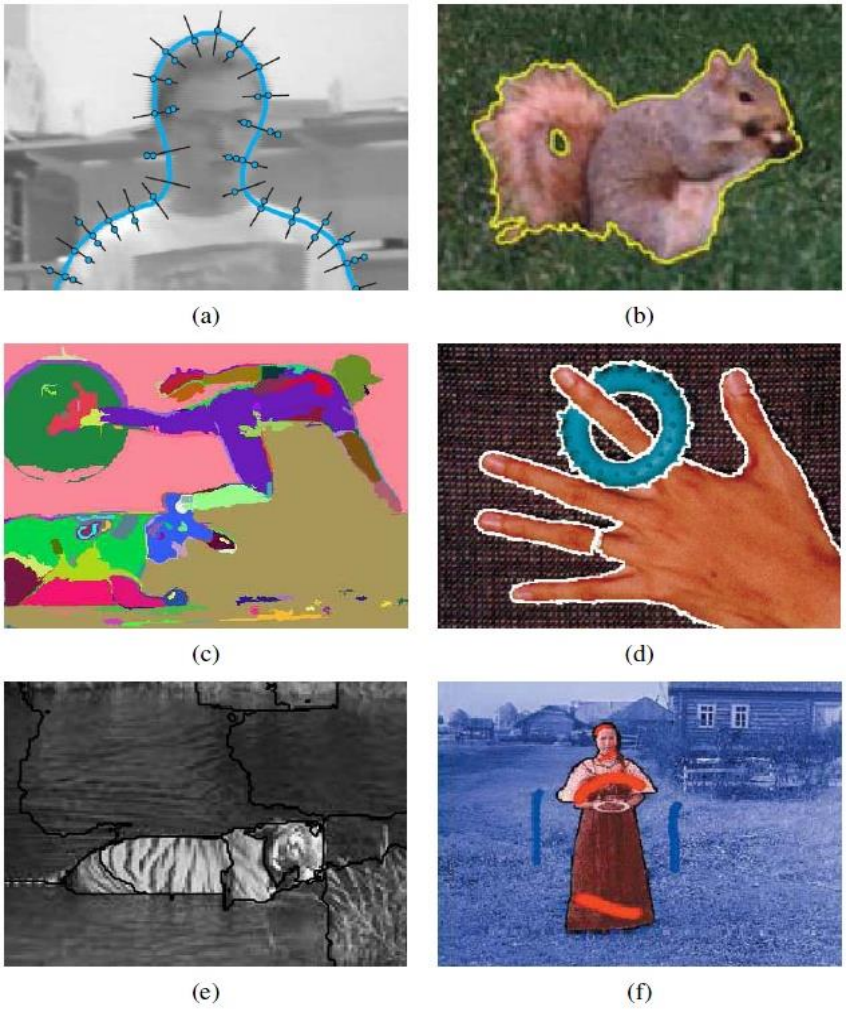
- Even within an application domain images may vary widely



All these examples are microscope images of biological cells

Introduction

- Results from several popular segmentation techniques
  a) Active contours
  b) Level sets
  c) Graph-based merging
  d) Mean shift
  e) Normalised cuts
  f) Binary MRF



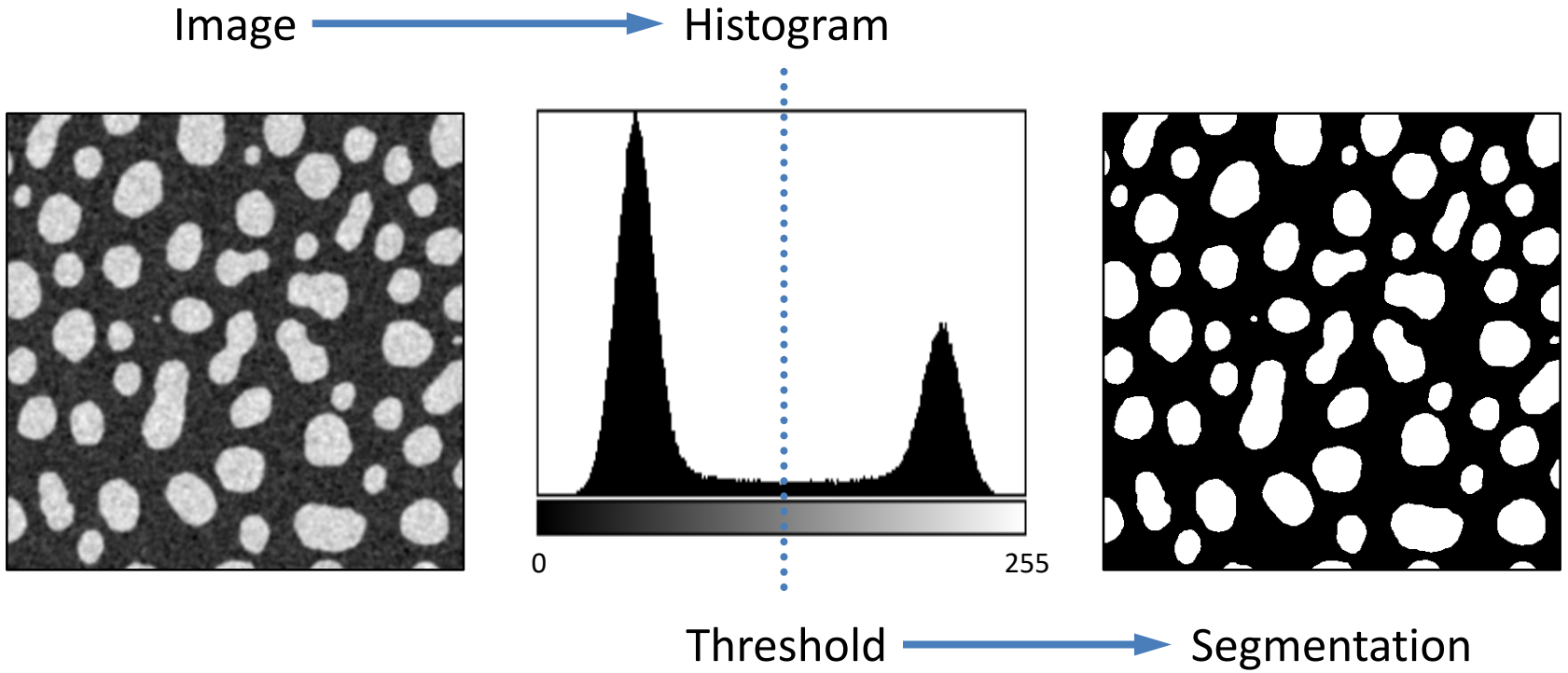
Outline

- Recap of basic segmentation methods
  - Thresholding
  - K-means clustering
  - Feature extraction and classification
- More sophisticated segmentation methods
  - Region splitting and merging
  - Watershed segmentation
  - Maximally stable extremal regions
  - Mean-shifting algorithm
  - Superpixel segmentation
  - Conditional random field
  - Active contour segmentation
  - Level-set segmentation
- How to evaluate segmentation methods
  - Quantitative evaluation metrics
  - Receiver operating characteristic

Recap of Basic Segmentation Methods

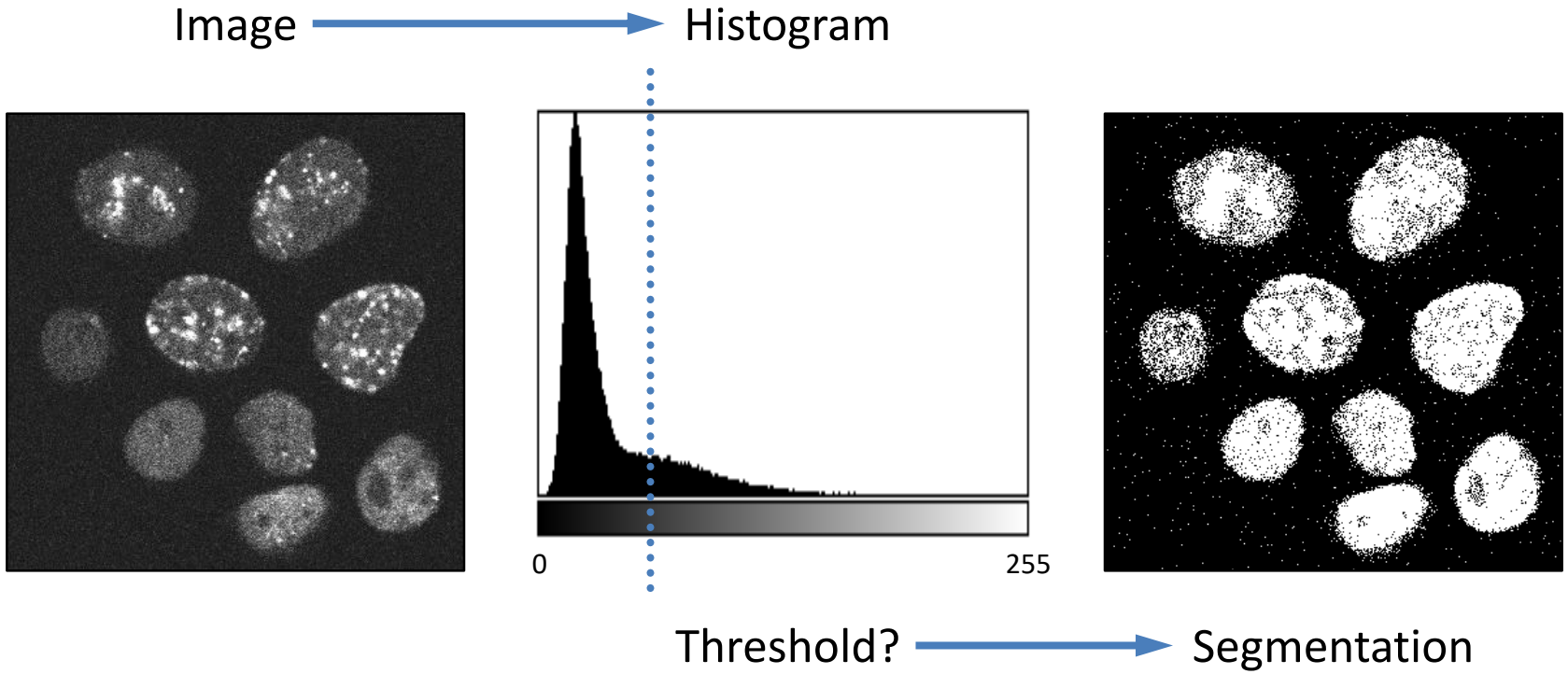
Thresholding

- Fine if regions have sufficiently different intensity distributions



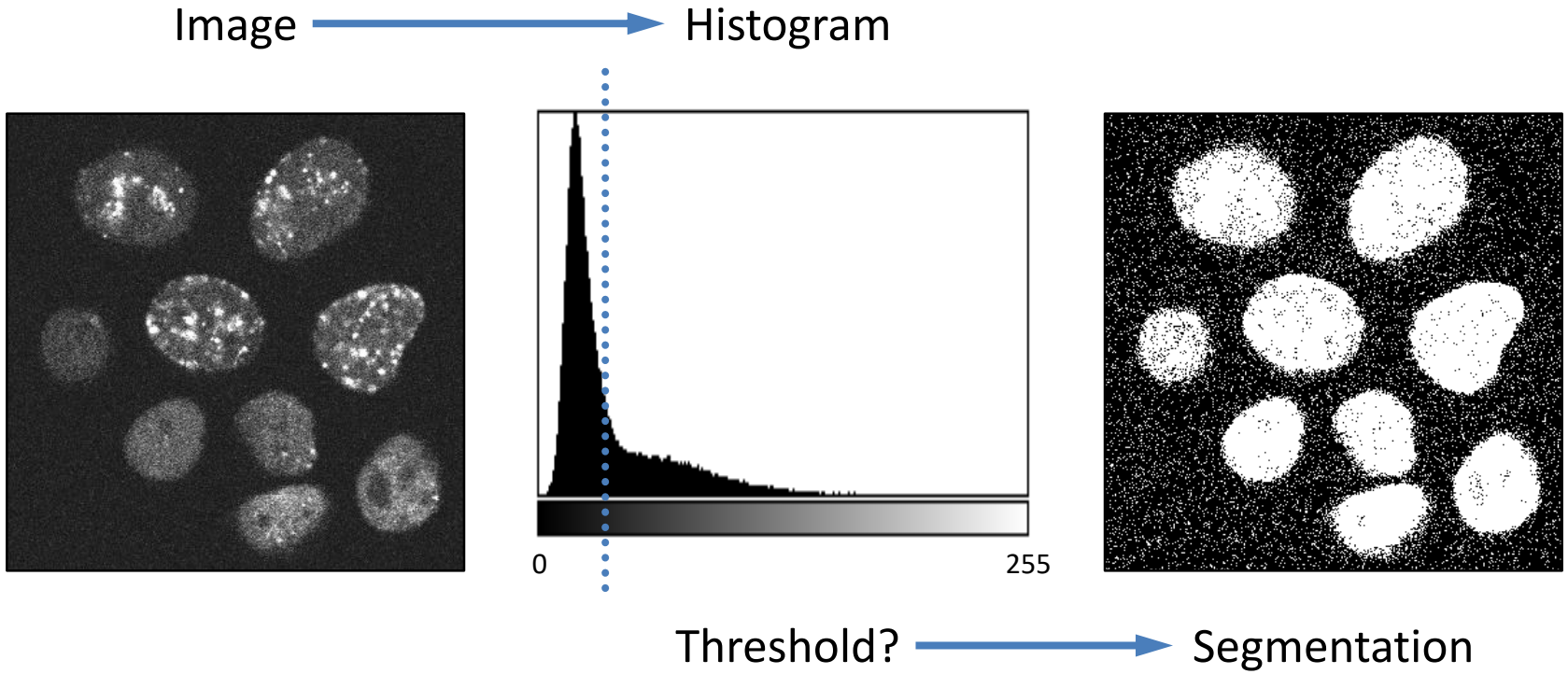
Thresholding

- Problematic if regions have overlapping intensity distributions



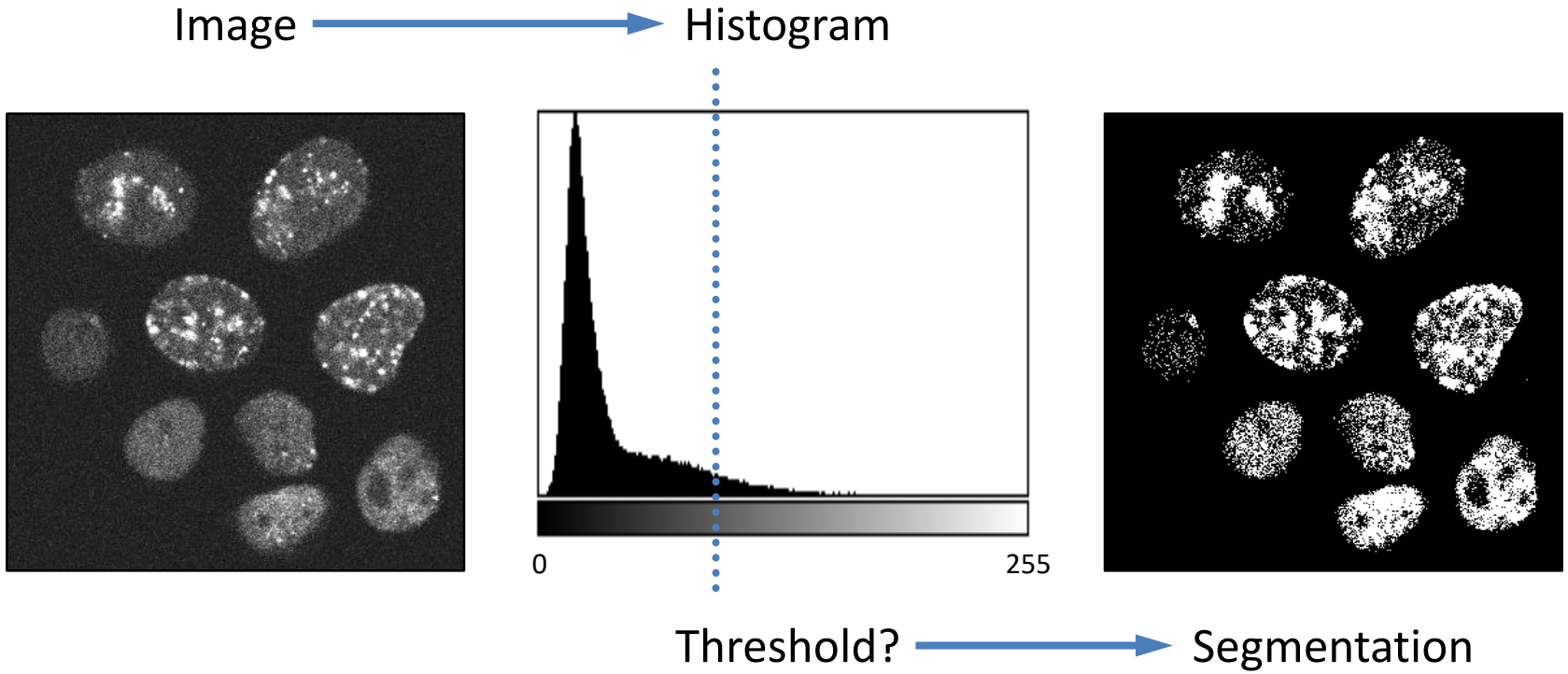
Thresholding

- Problematic if regions have overlapping intensity distributions



Thresholding

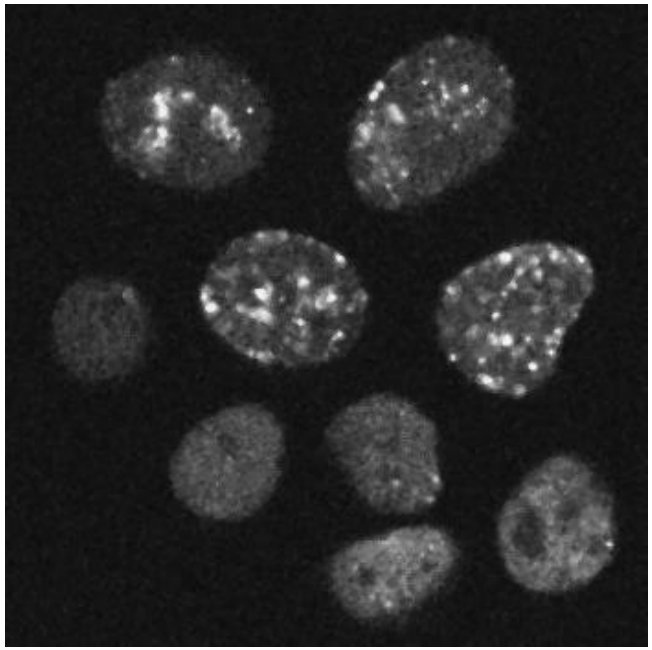
- Problematic if regions have overlapping intensity distributions



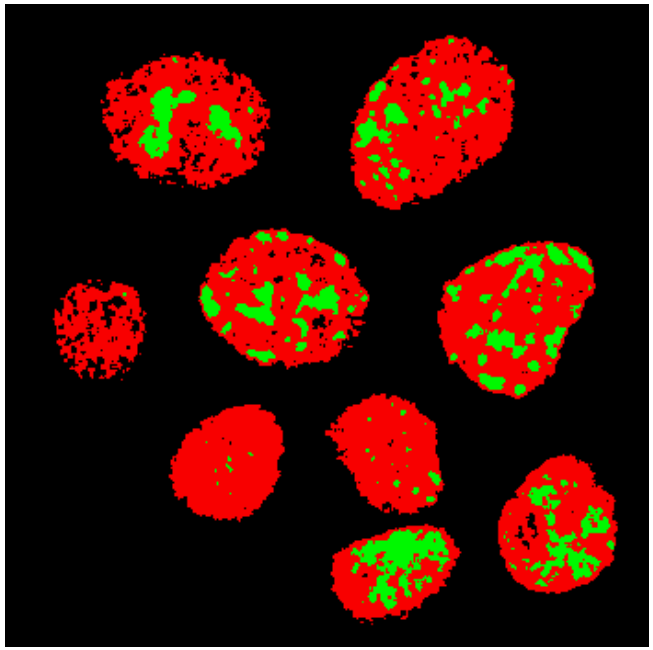
K-Means Clustering

- Could work if the number of clusters is known a priori

Original Image



Labeled Image



K-Means Clustering

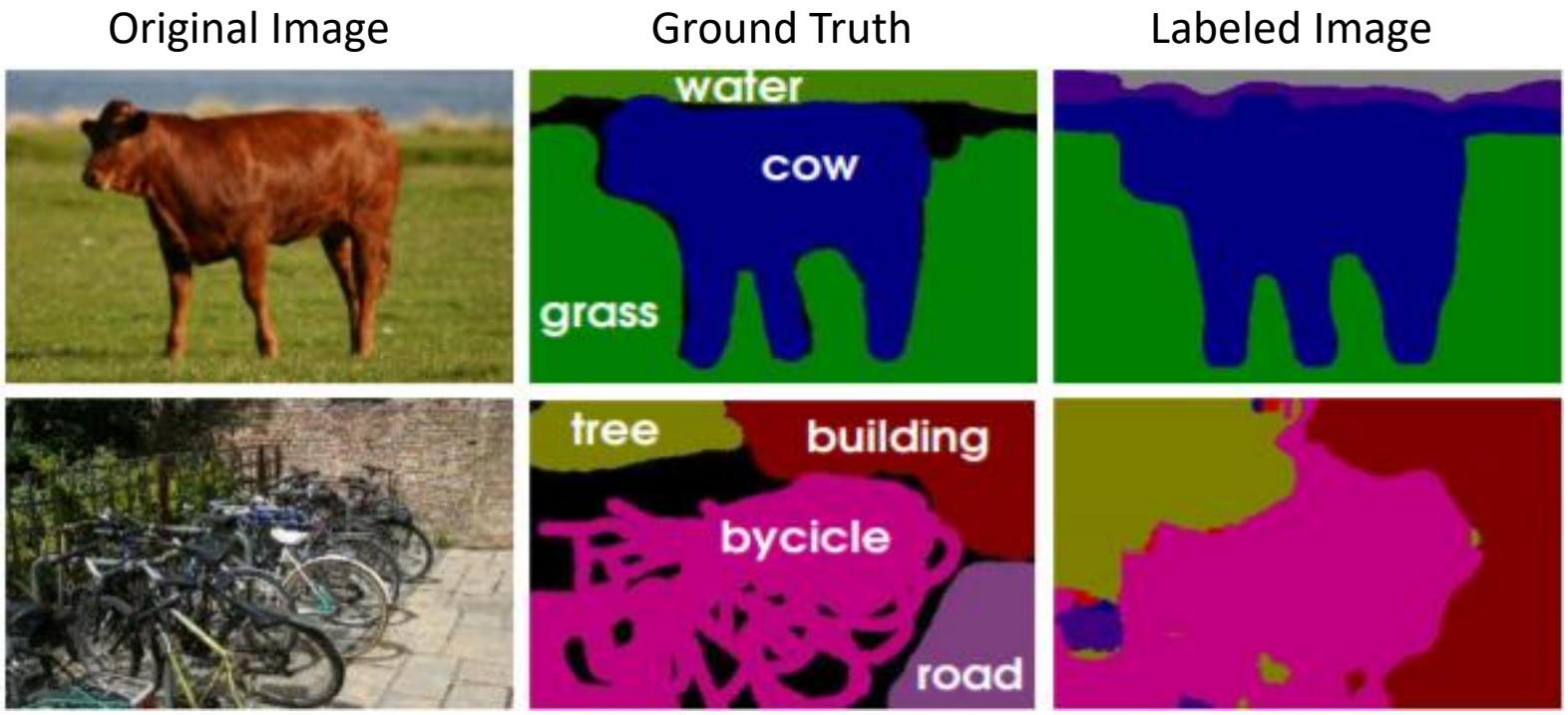
- Problematic if the number of clusters is not known a priori

Original Image

Labeled Image

Feature Based Classification

- Segmentation by sliding-window patch-wise feature extraction and then classification… requires many examples for training



Schroff et al. Single-histogram class models for image segmentation. In: Computer Vision, Graphics and Image Processing, Springer, 2006.

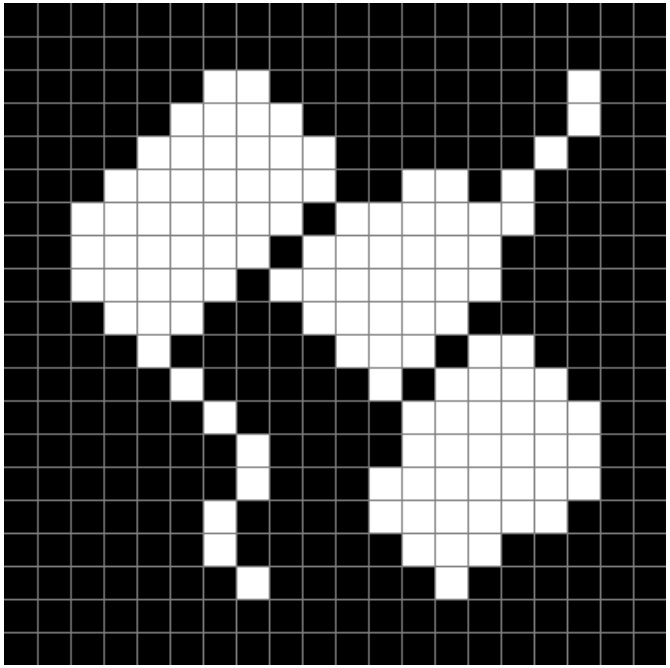
More Sophisticated Segmentation Methods

Region Splitting and Merging

- Basic computational approach
  - Recursively split the whole image into pieces based on region statistics
  - Recursively merge regions together in a hierarchical fashion
  - Combine splitting and merging sequentially

Region Splitting and Merging

- The simplest possible technique
  - Apply thresholding and then compute connected components
  - Rarely sufficient due to lighting and intra-object statistical variations

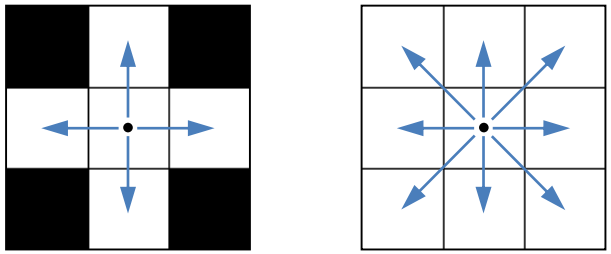


$$\theta(f,t) = \begin{cases} 1 & \text{if } f \geq t \\ 0 & \text{else} \end{cases}$$

How many connected components (separate objects) are there in this thresholded image?

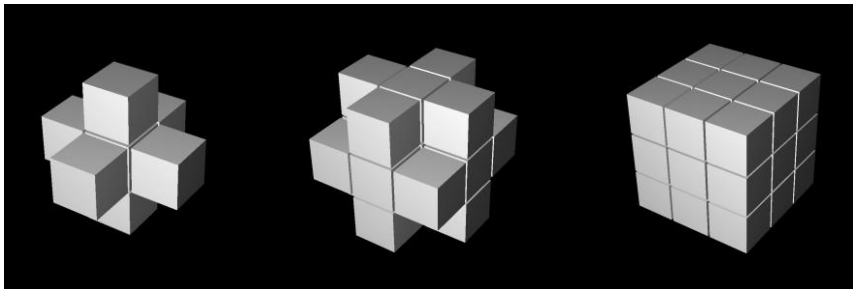
Connected Components

- Connectivity in two dimensions (2D)



4-connected 8-connected

- Connectivity in three dimensions (3D)

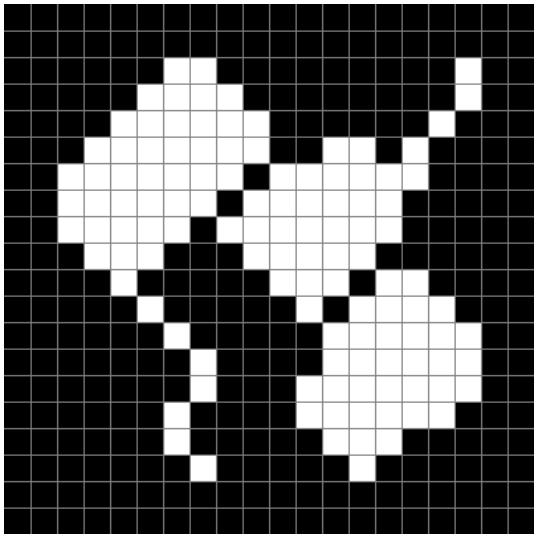


6-connected 18-connected 26-connected

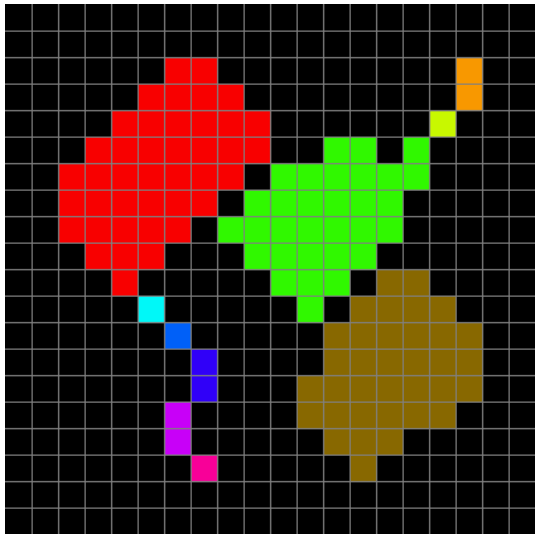
Connected Components

- Number of components depends on the chosen connectivity

Thresholded image

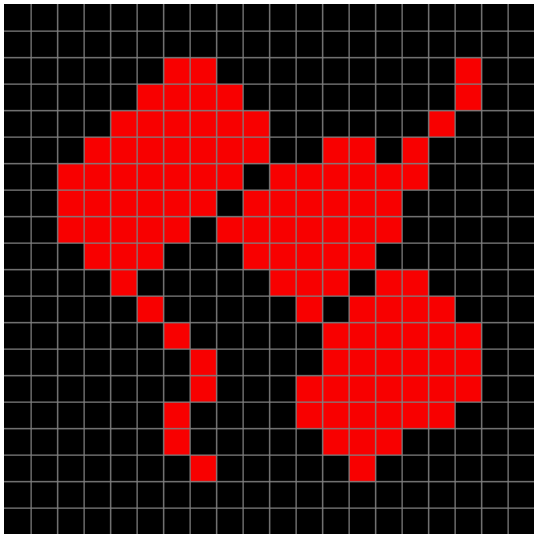


4-connected objects



Number of objects = 10

8-connected objects



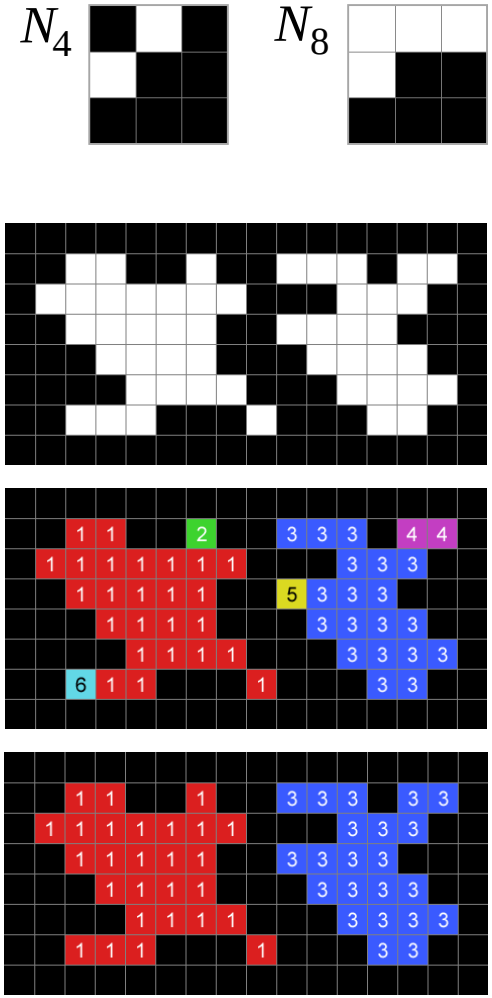
Number of objects = 1

Connected Components Algorithm

- First pass
  - Check each pixel (top-left to bottom-right)
  - If an object pixel, check its neighbors (N_4 or N_8)
  - If no neighbors have labels, assign a new label
  - If neighbors do have labels, assign the smallest
  - Record label equivalences while assigning

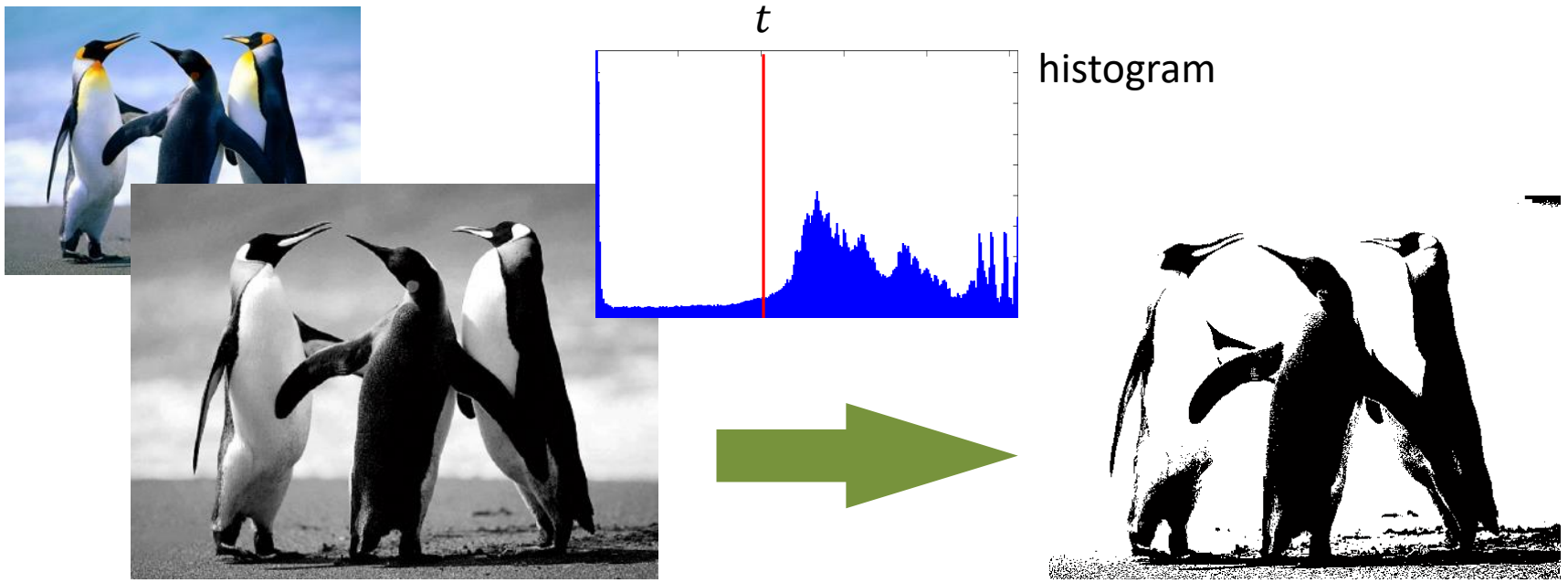
    Equivalence sets {1,2,6} {3,4,5}

- Second pass
  - Check each pixel (top-left to bottom-right)
  - Replace each label with its smallest equivalent
  - All background pixels default to the zero-label



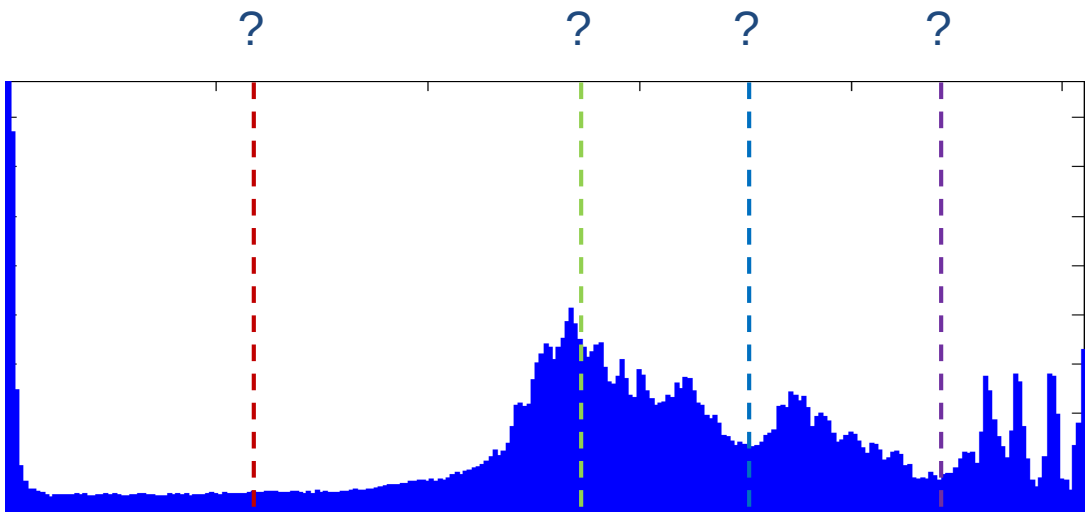
Region Splitting

- Basic computational approach
  - One of the oldest techniques in computer vision
  - First compute the histogram for the whole image
  - Then find a threshold t that best separates the peaks in the histogram



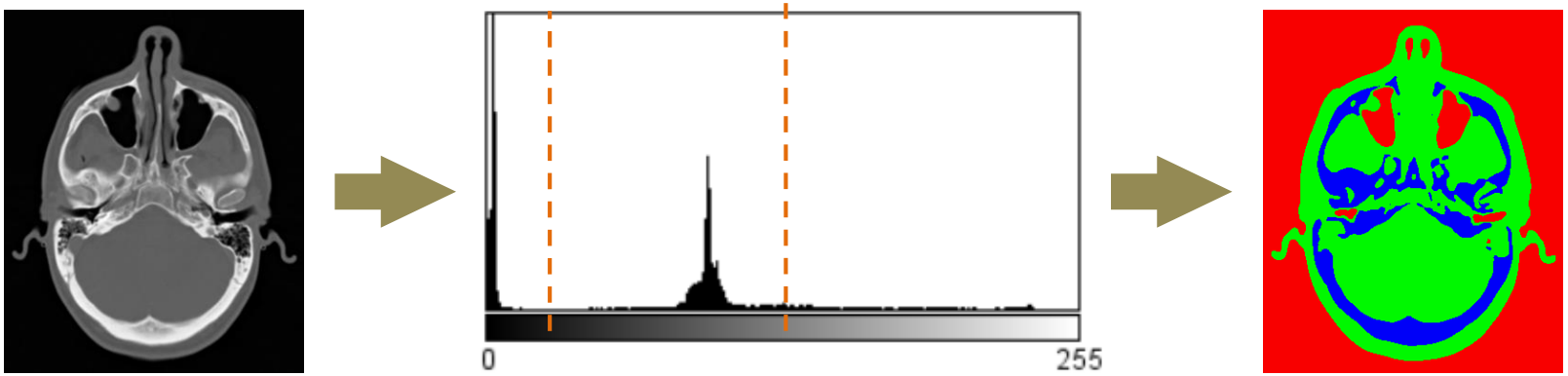
Region Splitting

- Basic computational approach
  - Repeat until regions are either fairly uniform or below a certain size
  - Optimise metrics of **intra-region similarity** and **inter-region dissimilarity**



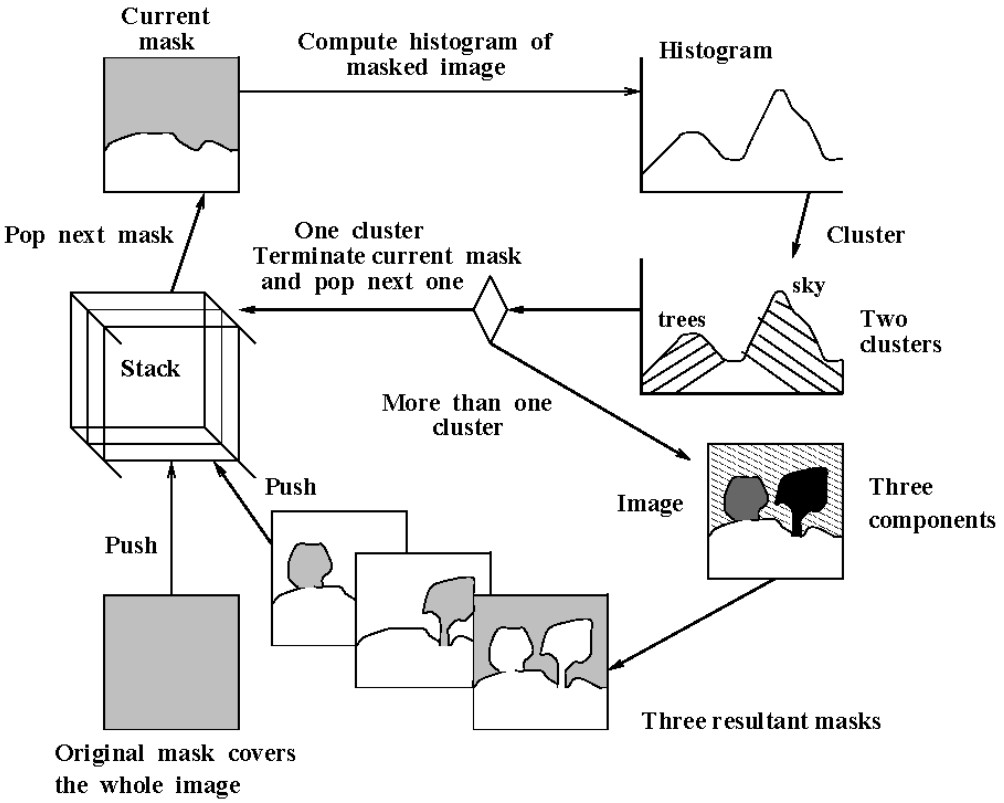
Region Splitting

- Histogram based methods
  - Follow a measurement space clustering process
  - Assume homogeneous objects in the image appear as clusters in measurement space (the histogram in our example)
  - Histogram clustering can be accomplished by finding the valleys and declaring the intervals between them as the clusters
  - Segmentation = mapping the clusters back to the image domain
  - Connected components of the cluster labels constitute the segments



Region Splitting

- Recursive version of histogram based splitting
  - Perform histogram mode seeking first on the whole image
  - Then on each of the regions obtained from the resultant clusters
  - Until the regions obtained cannot be decomposed further



Region Merging

- Heuristics-based region merging
  - Using the relative **boundary lengths** and **edge strengths**
  - Using the distance between the **closest points** or **farthest points**
  - Using the average **colour difference** or **size difference**

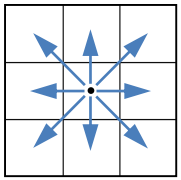
Region Merging

- Graph-based region merging
  - Use relative dissimilarities between regions R_i to merge
  - Represent regions as a graph using minimum spanning tree (MST)
  - Define a dissimilarity measure ω such as intensity difference
  - Compute internal region dissimilarity using the graph edges e

$$I(R) = \max_{e \in \mathrm{MST}(R)} \omega(e)$$

  - Compute dissimilarity between adjacent regions

$$D(R_i, R_j) = \min_{e \in (v_i \in R_i, v_j \in R_j)} \omega(e)$$



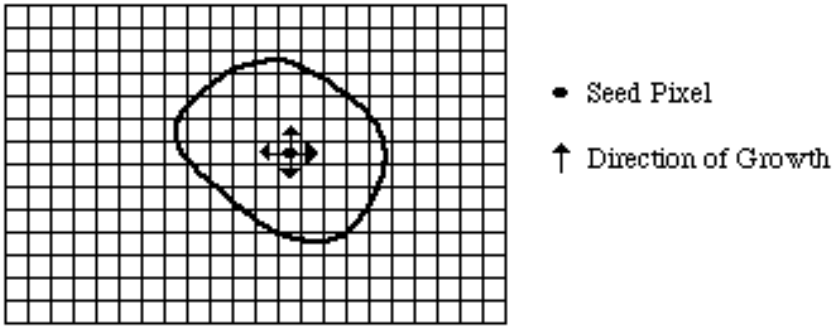
  - Merge any two adjacent regions whose dissimilarity is smaller than the minimum of the internal difference of these two regions

Region Merging

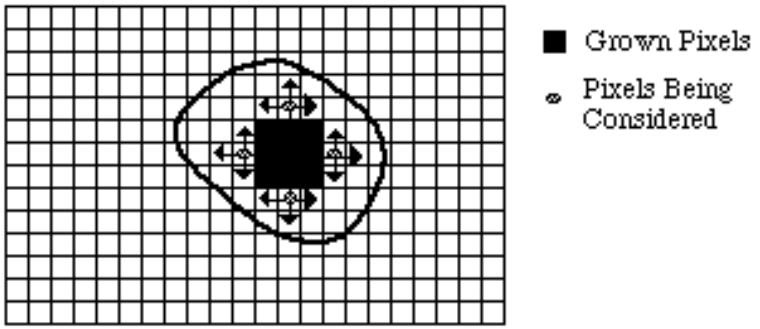
Felzenszwalb & Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision 59(2): 167–181, 2004.

Region Merging

- Merging by region growing
  - Define a similarity measure
  - Start from one seed pixel for the region
  - Add neighbouring pixels to the region if they are similar
  - Repeat previous step until no more pixels are similar



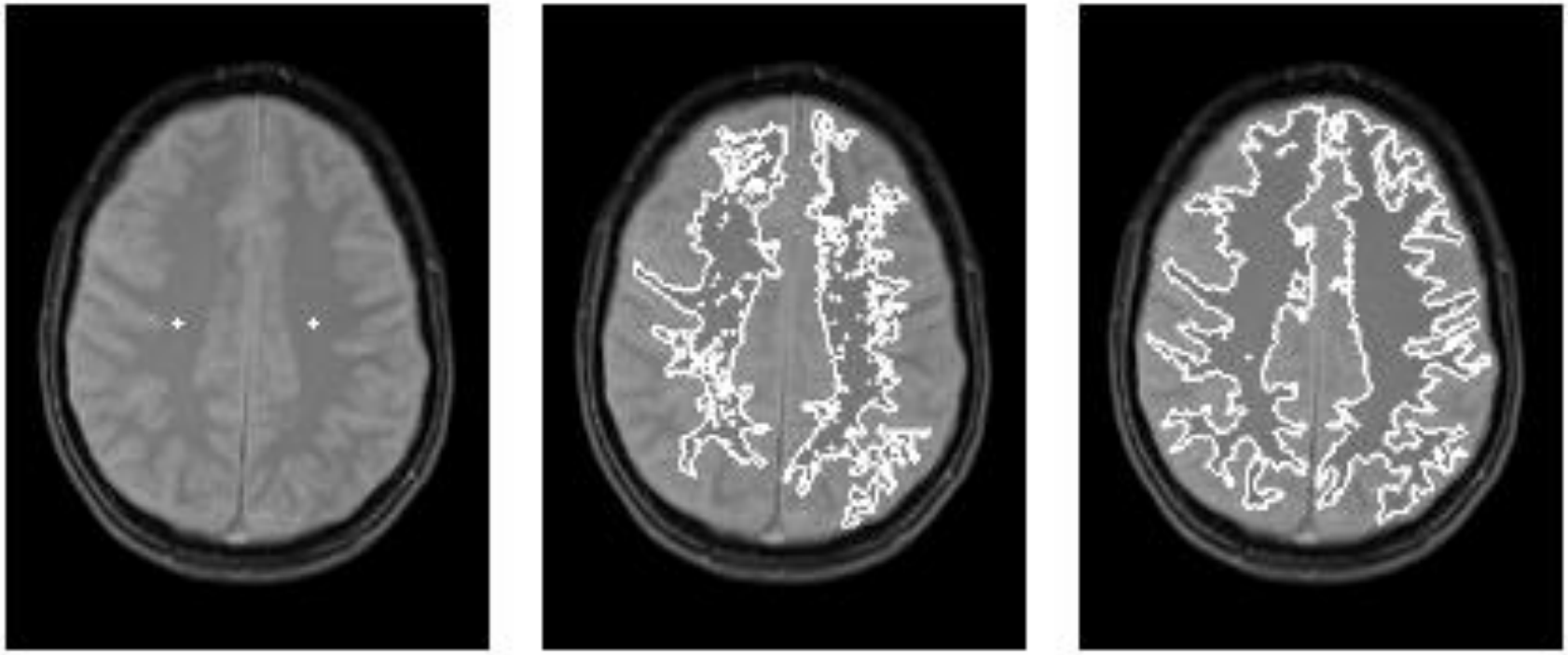
(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node35.html

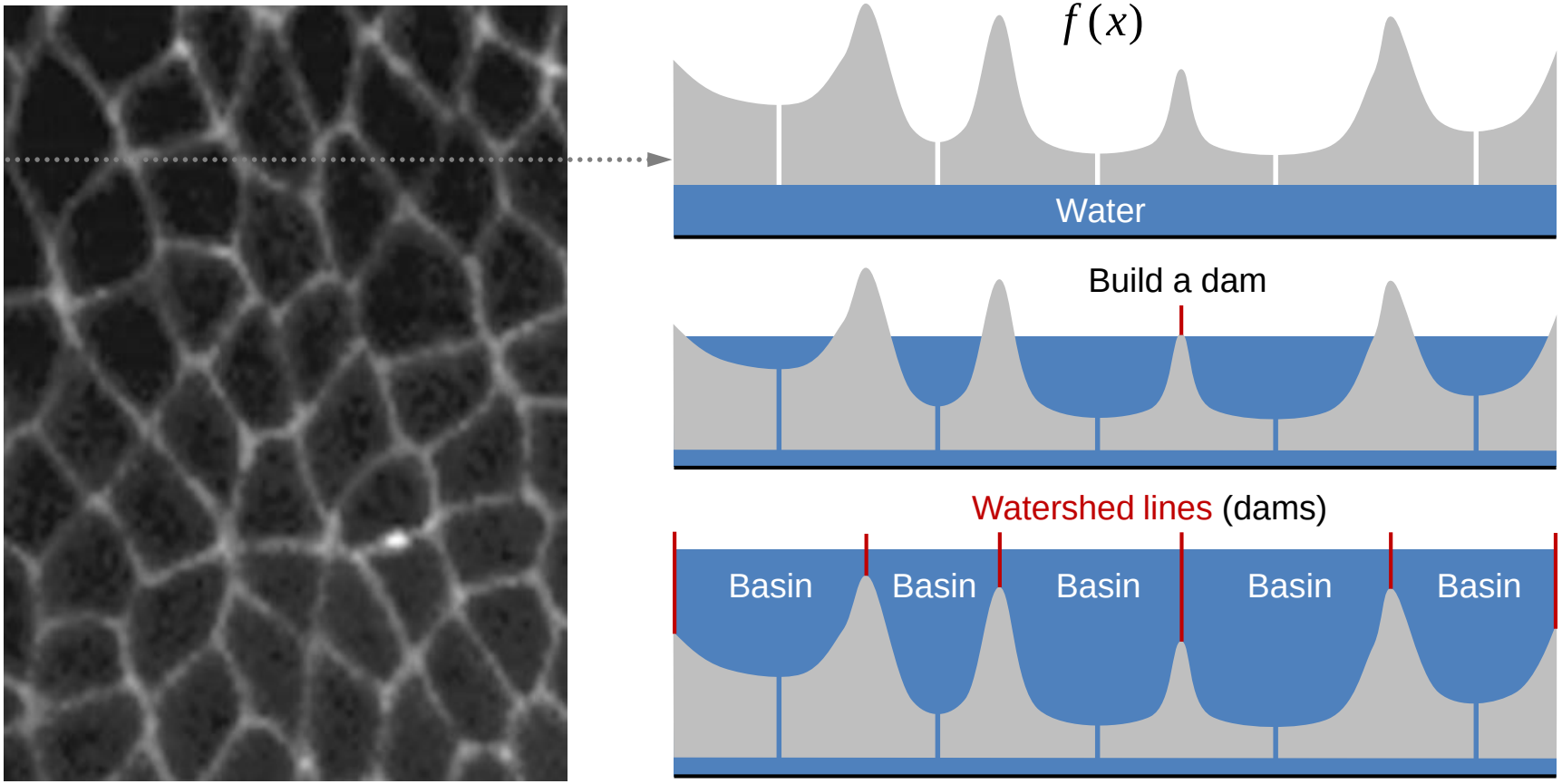
Region Growing



https://sbme-tutorials.github.io/2019/cv/notes/6_week6.html

Watershed Segmentation

- Based on the analogy of immersion of a topographic surface



Watershed Segmentation

- Meyer's flooding algorithm

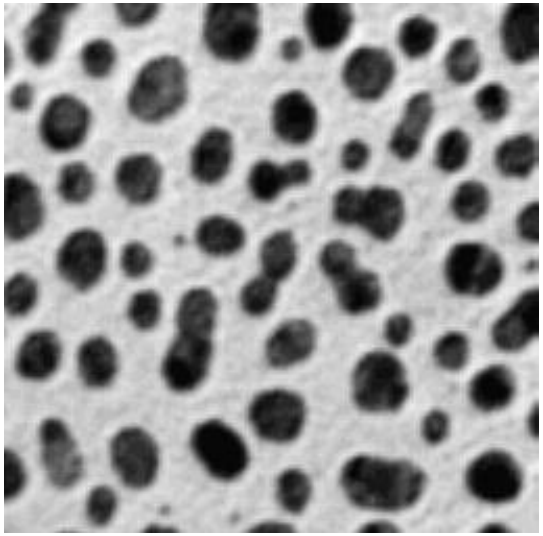
  1) Choose a set of markers to start the flooding. These could be, for example, all local minima. Give each a different label.
  2) Put neighbouring pixels of each marker into a priority queue with a priority level corresponding to the gray value of the pixel.
  3) Pop the pixel with the highest priority level from the queue. If the neighbours of the popped pixel that have already been labelled all have the same label, then give the pixel that same label. Put all non-labelled neighbours that have never been in the queue into the queue.
  4) Repeat step 3 until the queue is empty.

  The resulting non-labelled pixels are the watershed lines

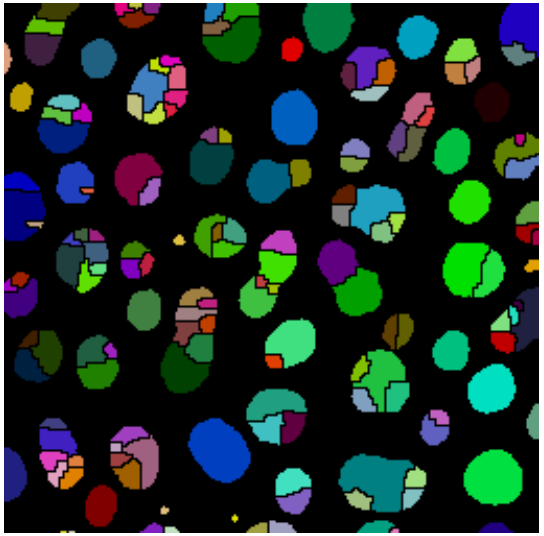
Watershed Segmentation

- Example segmentation results

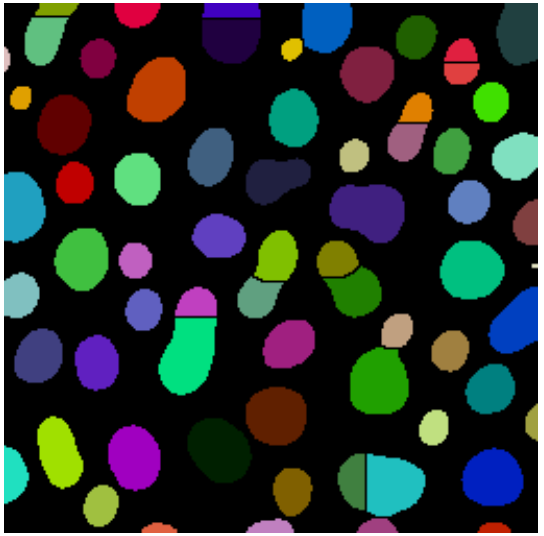
Input image



Segmentation results



Watershed of input

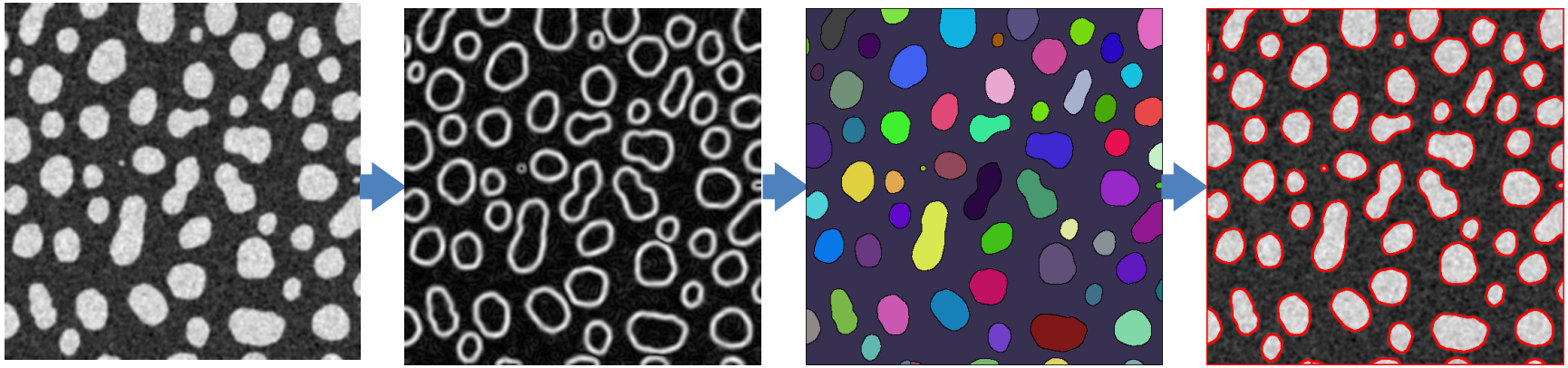


Watershed of smoothed input

https://imagej.net/Classic_Watershed

Watershed Segmentation

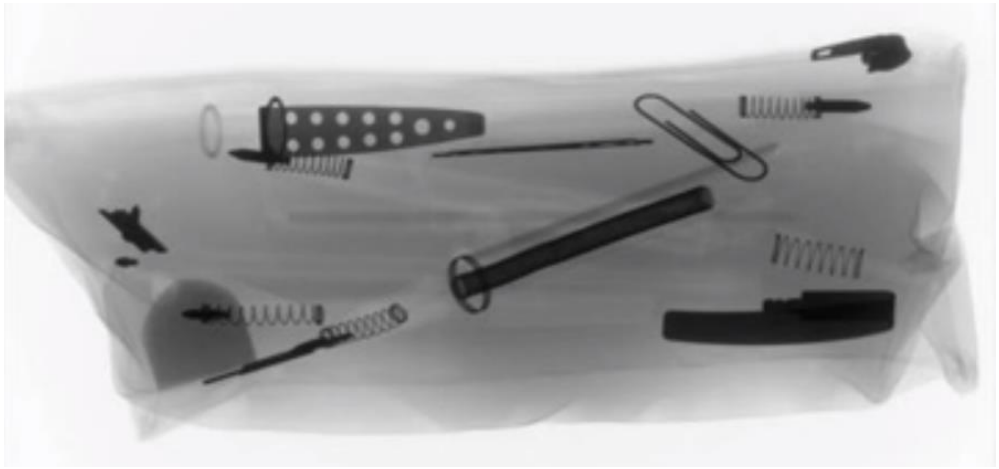
- Invert the image or compute edges if needed to get minima



- Important notes regarding watershed based segmentation
  - Images often have many local minima, leading to heavy oversegmentation
  - Preprocessing (image smoothing) may be needed to reduce false minima
  - Postprocessing (basin merging) may be needed to reduce fragmentation
  - Many different implementations and pre/postprocessing criteria exist
  - It is possible to start from user-defined markers instead of local minima

Maximally Stable Extremal Regions

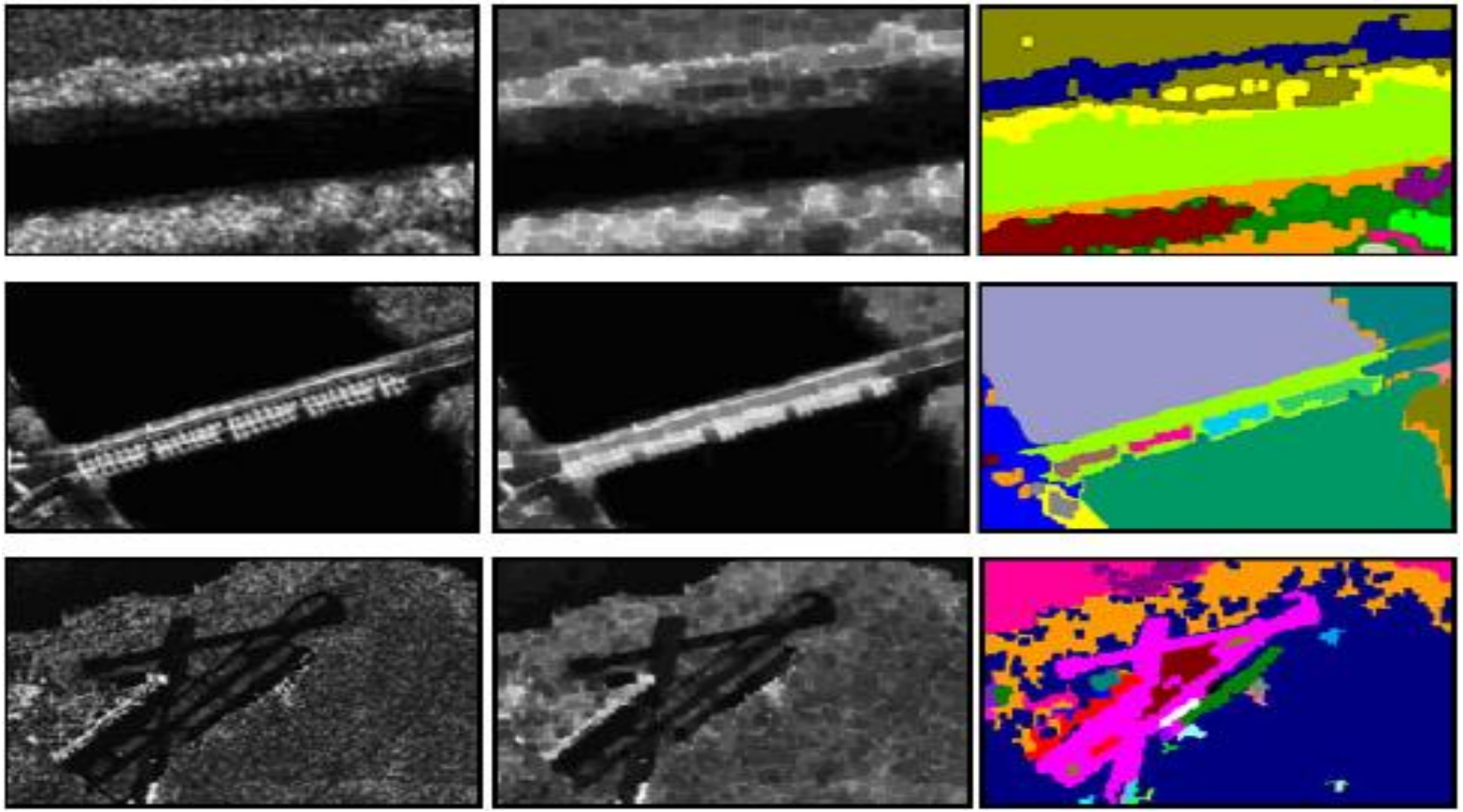
- Basics of maximally stable extremal regions (MSER)
  - Connected components characterised by almost uniform intensity surrounded by contrasting background
  - Constructed through a process of trying multiple thresholds and analyzing the shape of the connected components
  - Selected regions are connected components whose shape remains virtually unchanged over a large set of thresholds.



https://www.youtube.com/watch?v=tWdJ-NFE6vY

Maximally Stable Extremal Regions

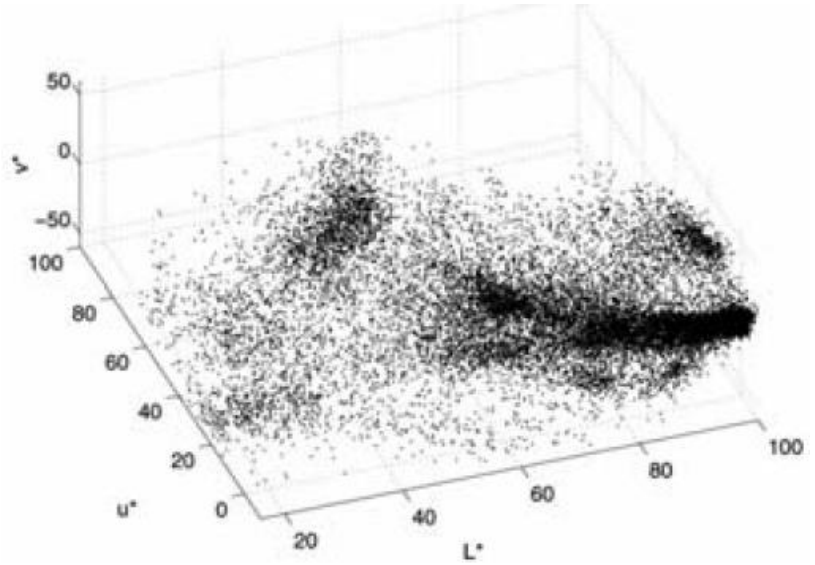
- Example MSER segmentation results



https://doi.org/10.1186/1687-6180-2012-83

Mean Shifting

- How would you segment this image based on colour alone?



Mean Shifting

- K-means clustering has limitations
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima

- Mean shifting is a good alternative in many applications
  - Seeks stationary points (peaks/modes) in a density function
  - Attempts to find all possible cluster centers in feature space
  - Does not require knowing the number of clusters a priori
  - Is a variant of iterative steepest-ascent method

Mean Shifting

- Iterative mode searching

1. Initialize a random seed point x and window N
2. Calculate the mean (center of gravity) $m(x)$ within N
3. Shift the search window to the mean
4. Repeat Step 2 until convergence

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

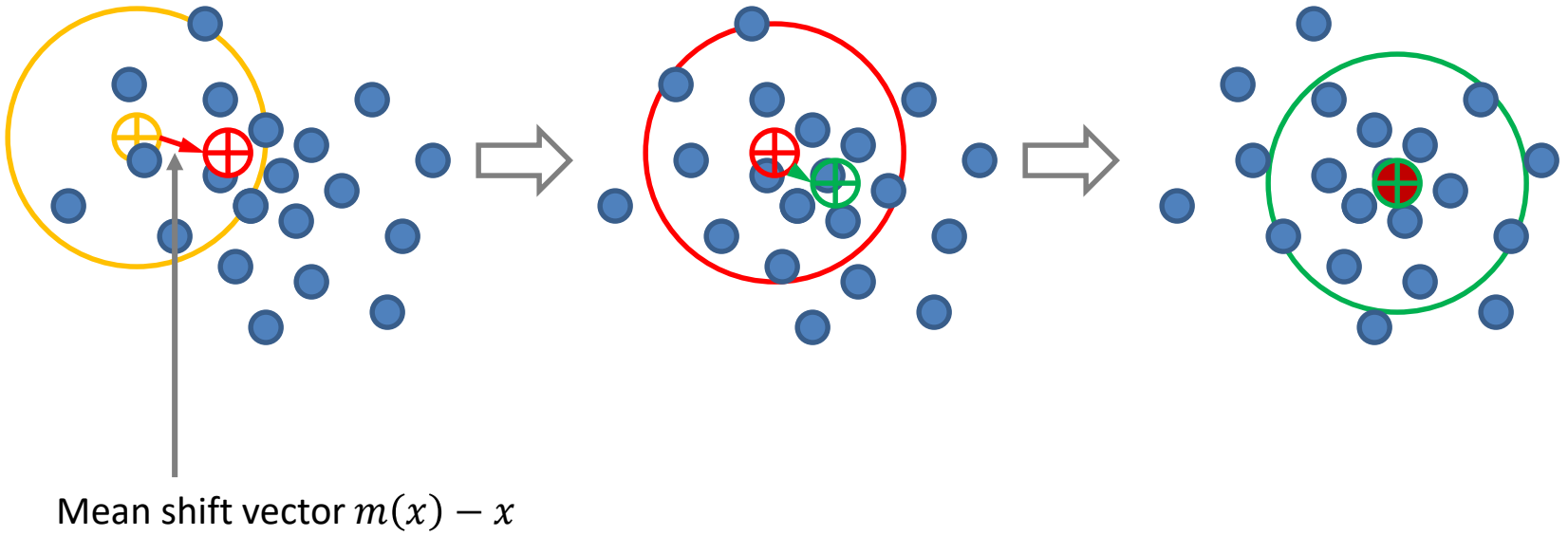
Mean (center of gravity)

$$K(x) = \exp(-|x|^2)$$

Kernel (weight function)

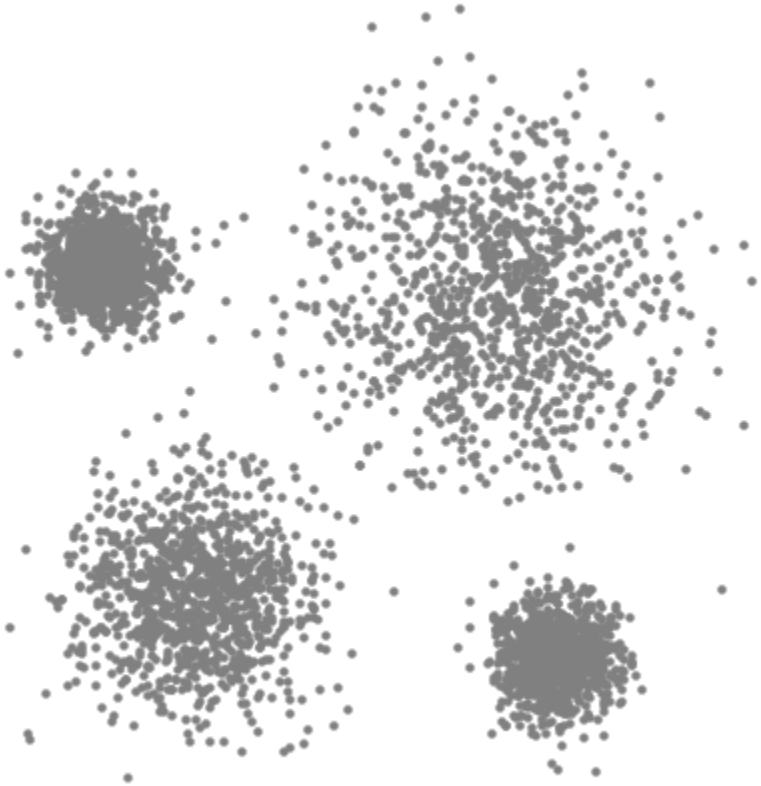
Mean Shifting

- Iterative re-estimation of the weighted mean



Mean Shifting

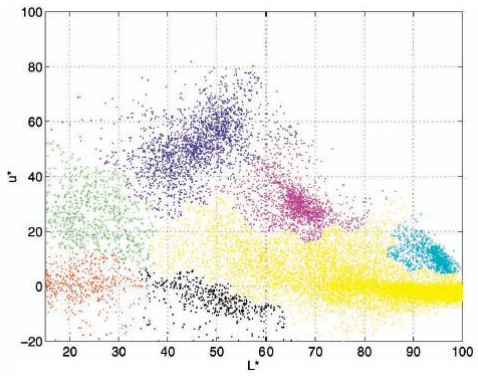
- Use a set of seed points to find all possible cluster centers

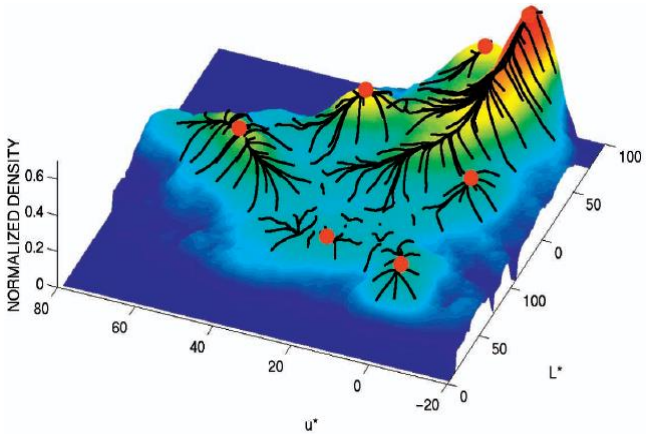


https://sbme-tutorials.github.io/2019/cv/notes/6_week6.html

Mean Shifting

- Performing image segmentation by mean shifting
  - Define features (colour, gradients, texture, et cetera)
  - Transform image pixels to points in the feature space
  - Initialise windows at multiple seed locations in feature space
  - Perform mean shifting for each window until convergence
  - Merge windows that end up near the same location
  - Cluster all points according to window traversal



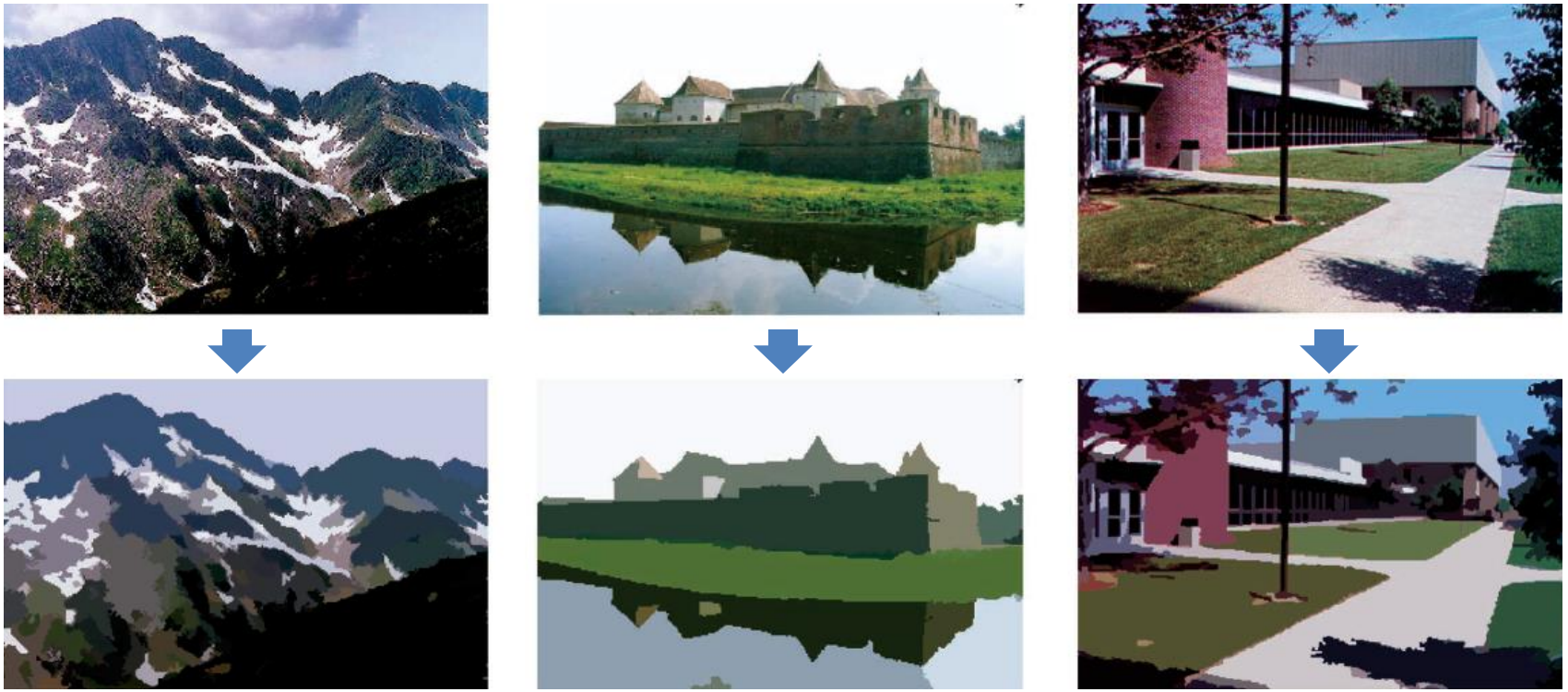


https://doi.org/10.1109/34.1000236

Mean Shifting

Mean Shifting

- Replace segmented region colours by their cluster means



https://doi.org/10.1109/34.1000236

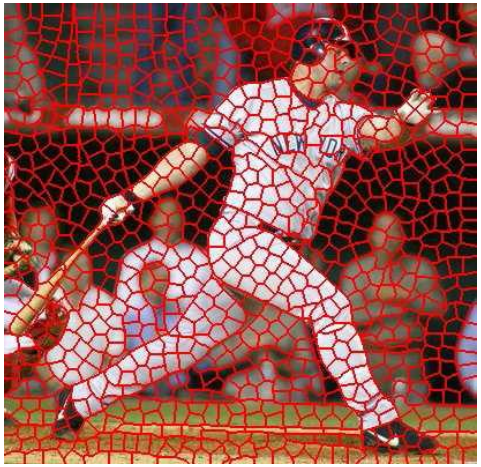
Mean Shifting

- Advantages
  - Model-free (does not assume any prior shape on data clusters)
  - Has just a single parameter (window size)
  - Finds variable number of modes (clusters)
  - Robust to outliers

- Limitations
  - Computationally expensive (need to shift many windows)
  - Output depends on window size parameter value
  - Window size (bandwidth) selection is not trivial
  - Does not scale well with dimensionality of feature space

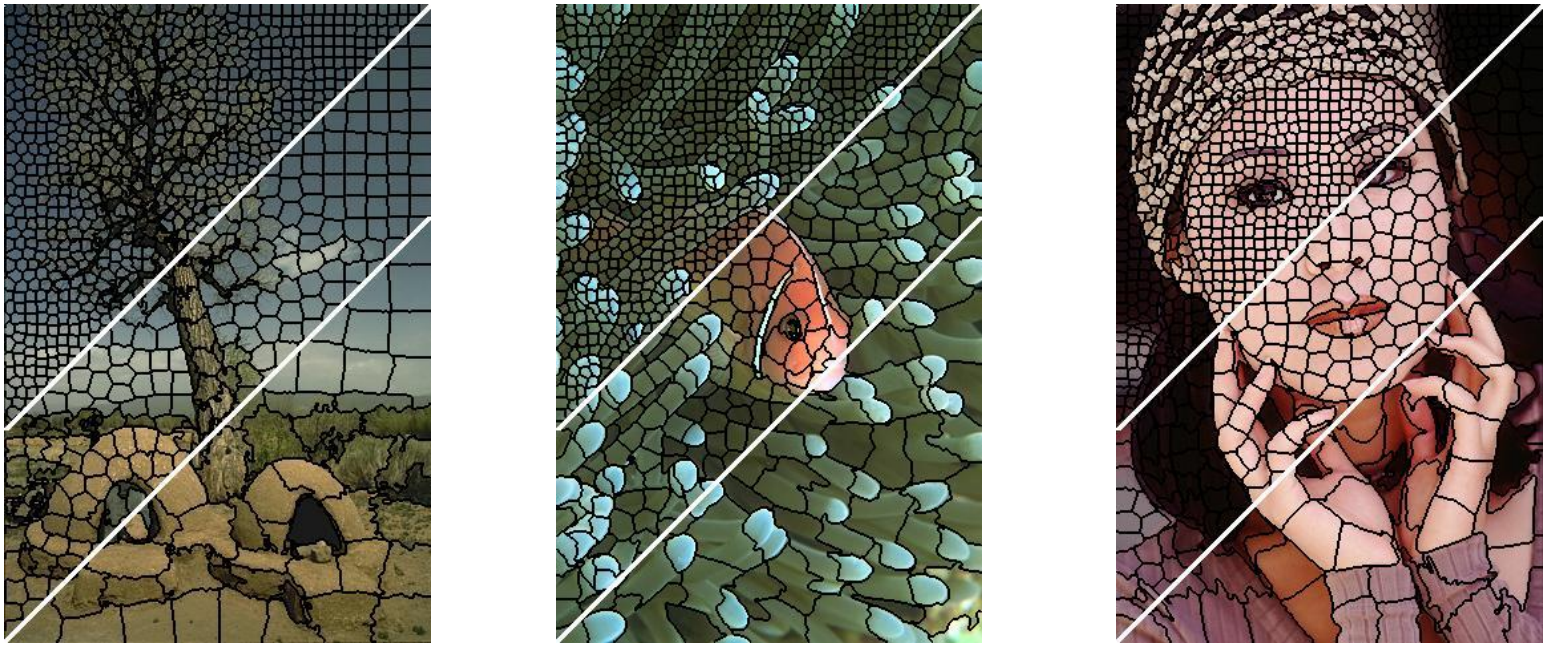
Superpixel Segmentation

- Pixel-wise sliding window segmentation
  - Too many windows (a.k.a. image patches)
- Superpixel-based segmentation improves efficiency
  - Group similar pixels into a superpixel
  - Superpixels together are an over-segmentation of the image
  - Ultimate segmentation (classification) performed on superpixels



Superpixel Segmentation

- Simple linear iterative clustering (SLIC)
  - A popular superpixel generation algorithm
  - Preserves image boundaries, is fast, and memory efficient



Superpixels of size 64, 256, and 1024 pixels (approximately)
https://ivrl.epfl.ch/research-2/research-current/research-superpixels/

Superpixel Segmentation

- Simple linear iterative clustering (SLIC)
  1. Initialise cluster centers C_j on pixel grid with step size S
  2. Move C_j to position in 3×3 window with smallest gradient
  3. Compute distance D_{ij} for each pixel i in $2S \times 2S$ window around C_j
  4. Assign each pixel i to the cluster C_j with smallest distance D_{ij}
  5. Recompute cluster centers as mean colour and position of pixels in C_j
  6. Iterate (go to Step 3) until the residual error is small

$$d_{lab} = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$ (distance in CIELAB color space)

$$d_{xy} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$ (distance in pixel space)

$$D = \sqrt{\frac{d_{lab}^2}{m^2} + \frac{d_{xy}^2}{S^2}}$$ (weight m controls influence of color over spatial distance)

 https://doi.org/10.1109/TPAMI.2012.120

Conditional Random Field

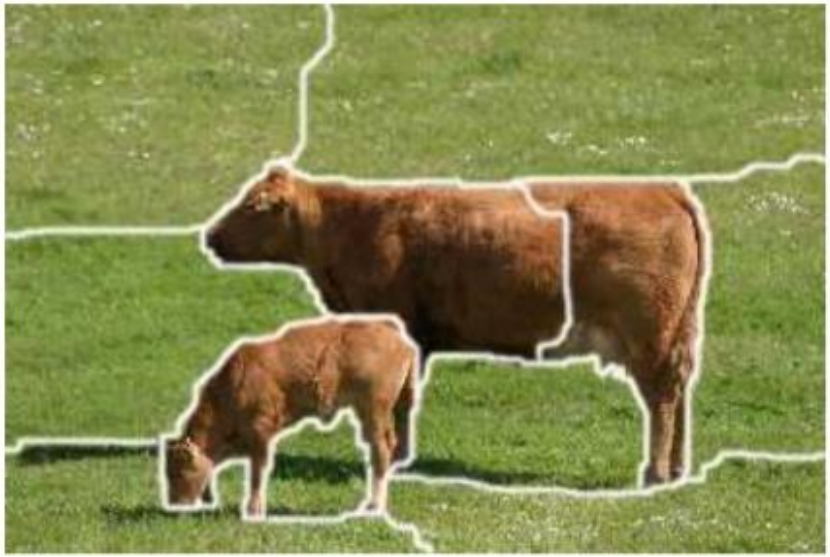
- Superpixels provide a basis for further segmentation
  - Determine spatial relationship between the superpixels
  - Compute similarities between superpixels
  - Group superpixels to form larger segments

- Conditional random field (CRF) approach

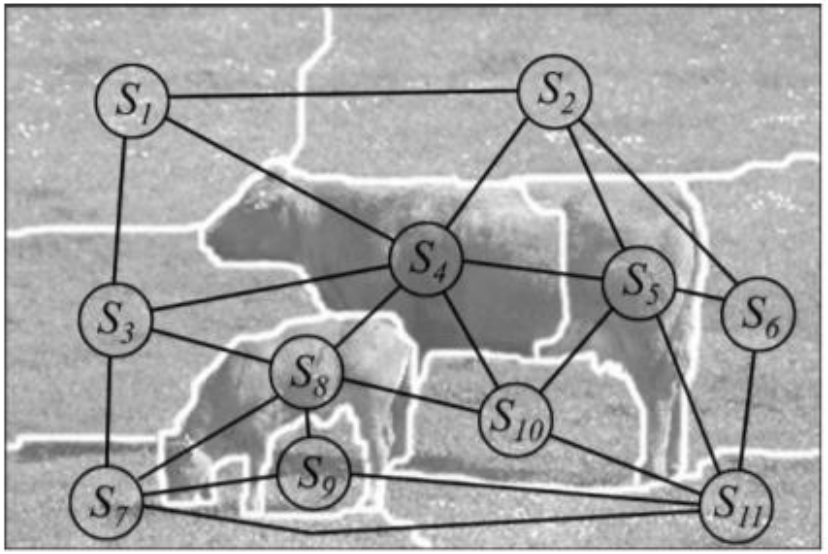
  A probabilistic graphical model that encodes the relationships between observations (i.e. superpixels) and constructs a consistent interpretation (i.e. segmentation) for a set of data (i.e. an image)

Conditional Random Field

- An undirected graphical structure
  - Nodes: superpixels (value based on features of superpixels)
  - Edges: adjacency (value based on similarity between superpixels)



Superpixels



Graph

https://doi.org/10.1007/s11263-008-0140-x

Conditional Random Field

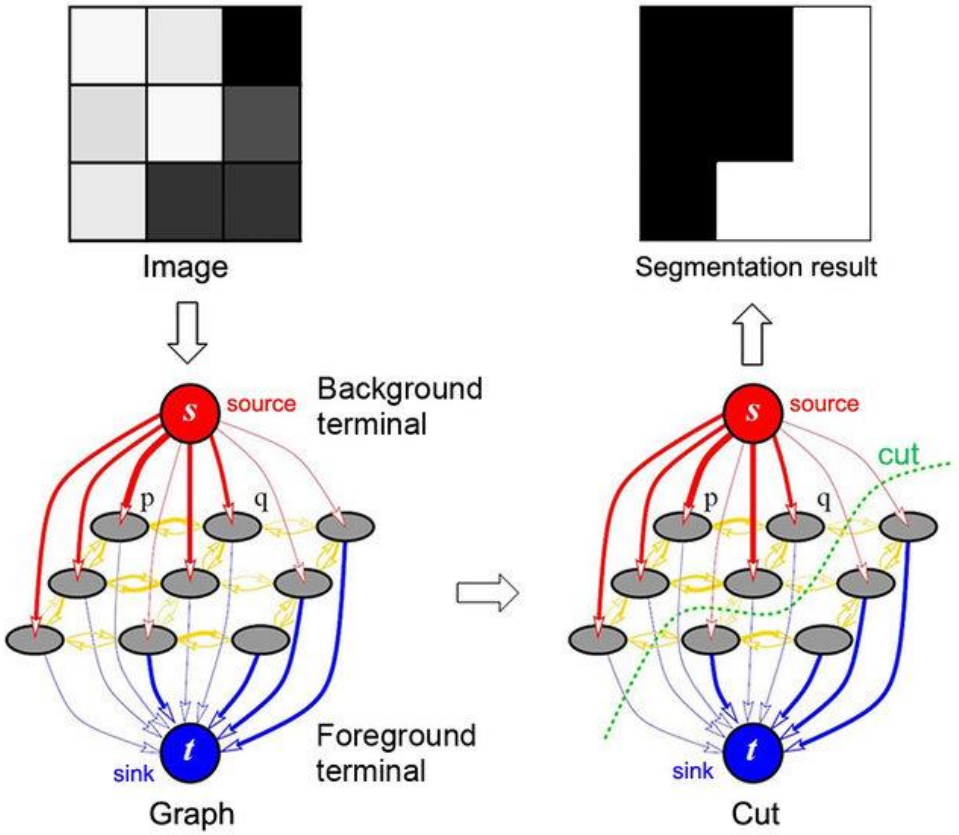
- Segmentation as an energy minimisation problem

$$E(s,c) = \sum_i \varphi(s_i, c_i) + \sum_{ij} \psi(s_i, s_j)$$

  - Unary potentials φ
    - Data term (based on graph node values)
    - Computes the cost of superpixel s_i belonging to class c_i
    - A lower cost means a higher likelihood of s_i belonging to c_i
    - Can be obtained via superpixel classification
  - Pairwise potentials ψ
    - Smoothness term (based on graph edge values)
    - Computes a cost of neighbourhood consistency
    - A cost is assigned if adjacent superpixels are assigned to different classes
    - Higher similarity results in a higher cost (0 if assigned to the same class)

Conditional Random Field

- Energy minimization is solved via *graph cut* (max-flow min-cut)



https://doi.org/10.1109/TPAMI.2004.60
https://doi.org/10.1109/TGRS.2016.2627638

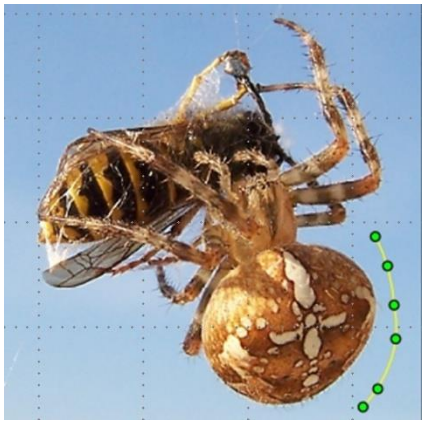
Conditional Random Field

- Segmentation example with multiple source-sink terminals

https://doi.org/10.1109/rivf.2012.6169870

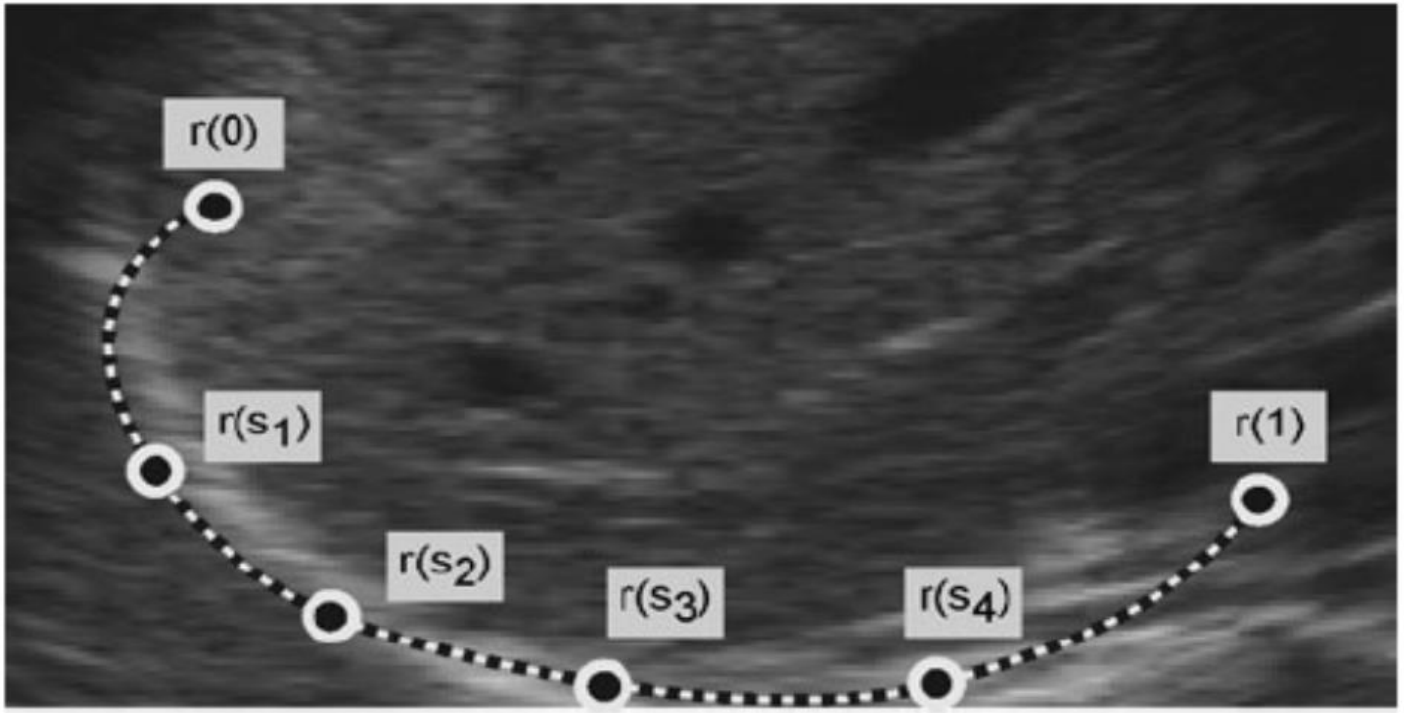
Active Contour Segmentation

- A contour based approach to object segmentation
  - Aims to locate object boundaries in images by curve fitting
  - Represents the curve by a set of control points and interpolation
  - Iteratively moves the control points to fit the curve to the object
  - Uses **image**, **smoothness** and **user-guidance forces** along curve



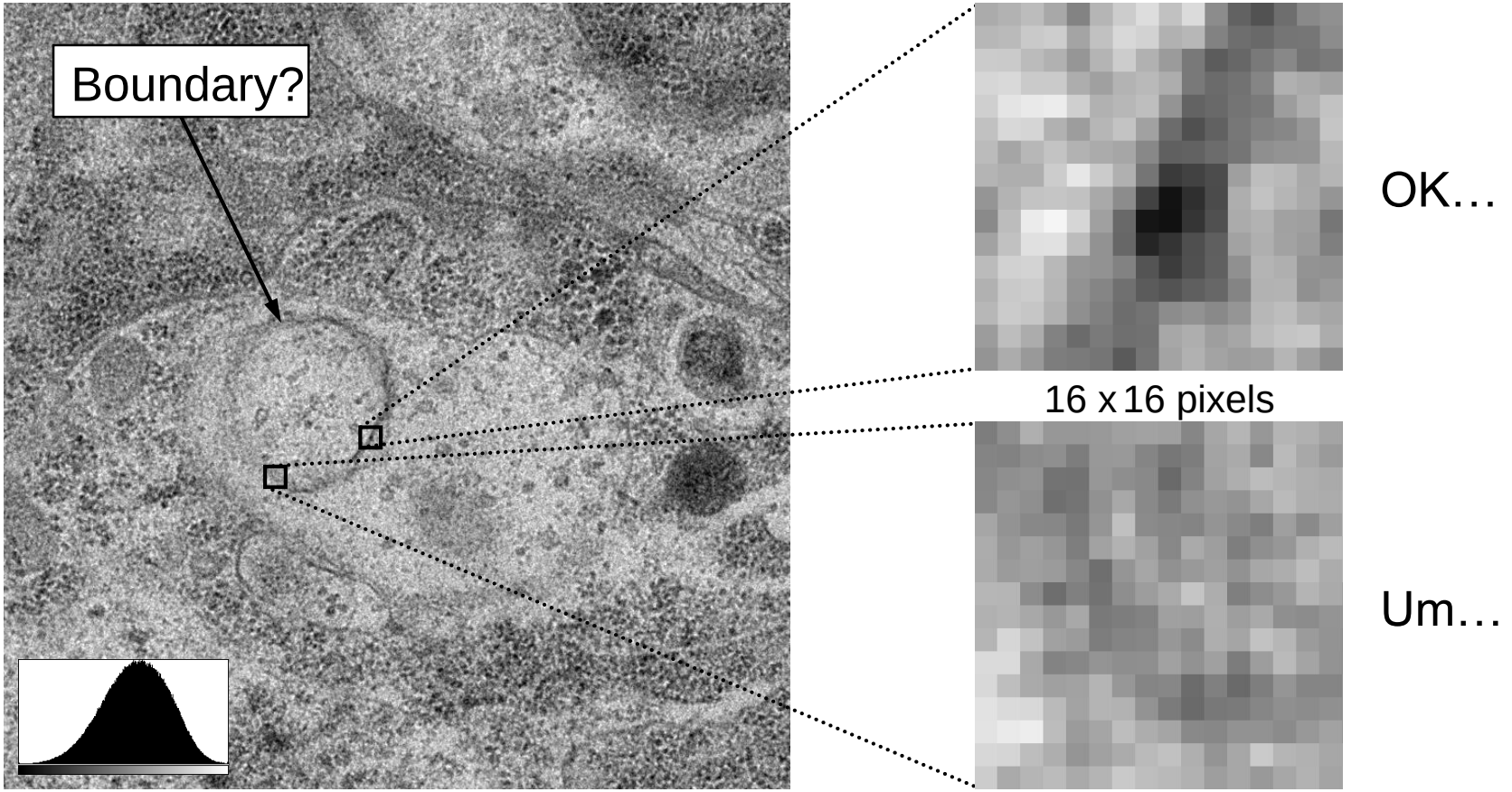

Active Contour Segmentation

- A well-known implementation is called the **snakes algorithm**
  - Smoothly follows high intensity gradients at the object boundary
  - Bridges areas of noise or missing gradients using smooth interpolation



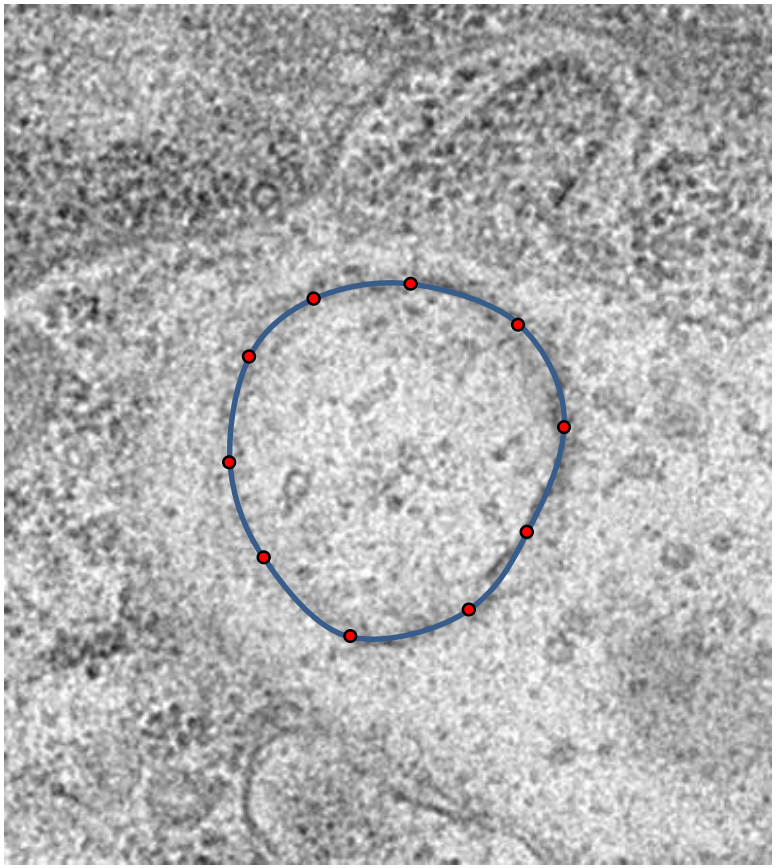
Active Contour Segmentation

- Enforces a solution where gradient information is lacking



Active Contour Segmentation

- Enforces a solution where gradient information is lacking



Define a suitable contour model

Circle, ellipse, or spline (open or closed)

$$C:[0,1]\to\mathbb{R}^n \underset{n=2}{\Rightarrow} C(s)=\left[x(s),y(s)\right]$$

Define a suitable energy functional

Using internal + image + external energies

$$E_C=\int_0^1\left[E_{\text{int}}\left(C(s)\right)+E_{\text{img}}\left(C(s)\right)+E_{\text{ext}}\left(C(s)\right)\right]ds$$

E.g. Bending energy; Intensity or gradient; User interaction

Minimize the energy functional

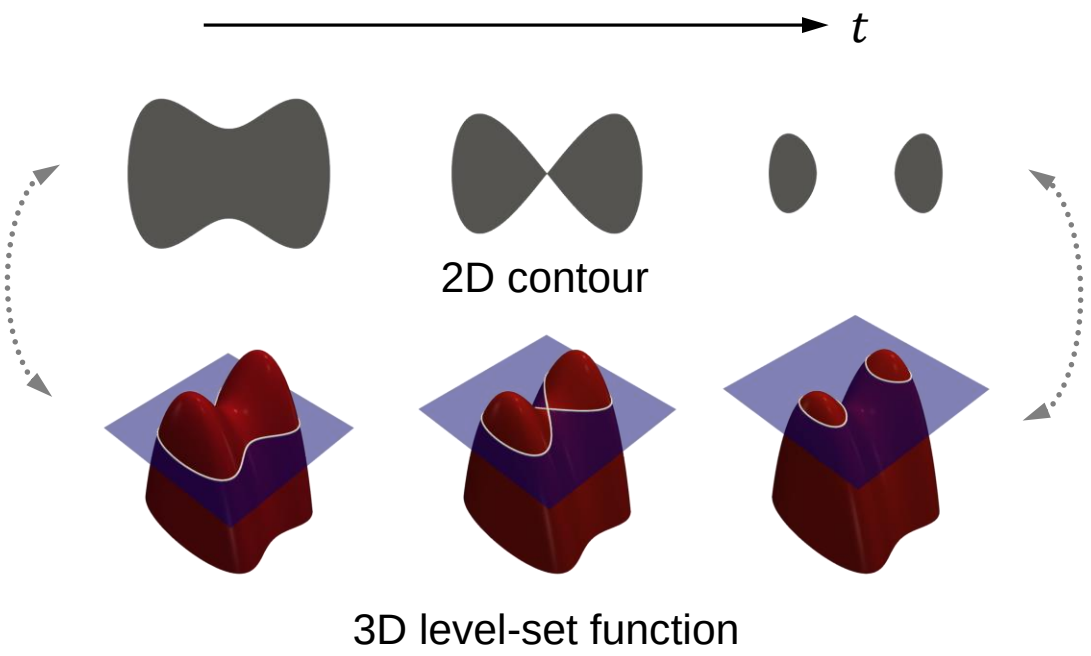
Using efficient optimization algorithms

Active Contour Segmentation

- Active contours / snakes are parametric models
  - Explicit representations of the object boundaries
  - Typically requires manual interaction to initialize the curve
  - It is challenging to change the topology of the curve as it evolves
  - Curve reparameterization may be required for big shape changes
- Level-set methods have become more popular alternatives
  - Implicit representations of the object boundaries
  - Boundaries defined by the zero-set of a higher dimensional function
  - Level-set function evolves to make the zero-set fit and track objects
  - Easily accommodates topological changes in object shape
  - Computationally more demanding than active contours

Level Set Segmentation

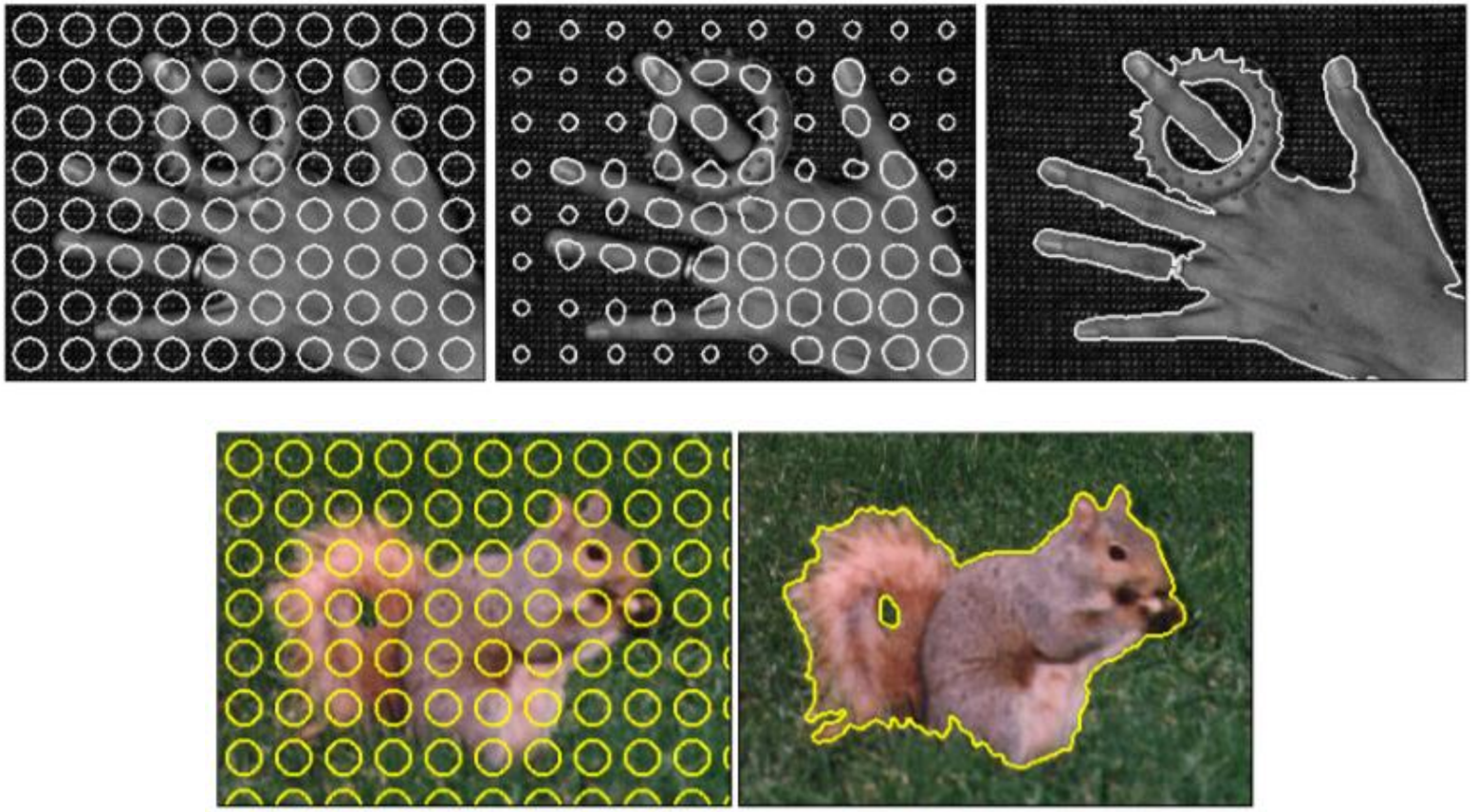
- Contour evolution over time (iterative)



https://en.wikipedia.org/wiki/Level-set_method

Level Set Segmentation

- Examples of level-set based segmentation



 https://doi.org/10.1007/s11263-006-8711-1

Level Set Segmentation

- A well-know implementation is the Chan-Vese model

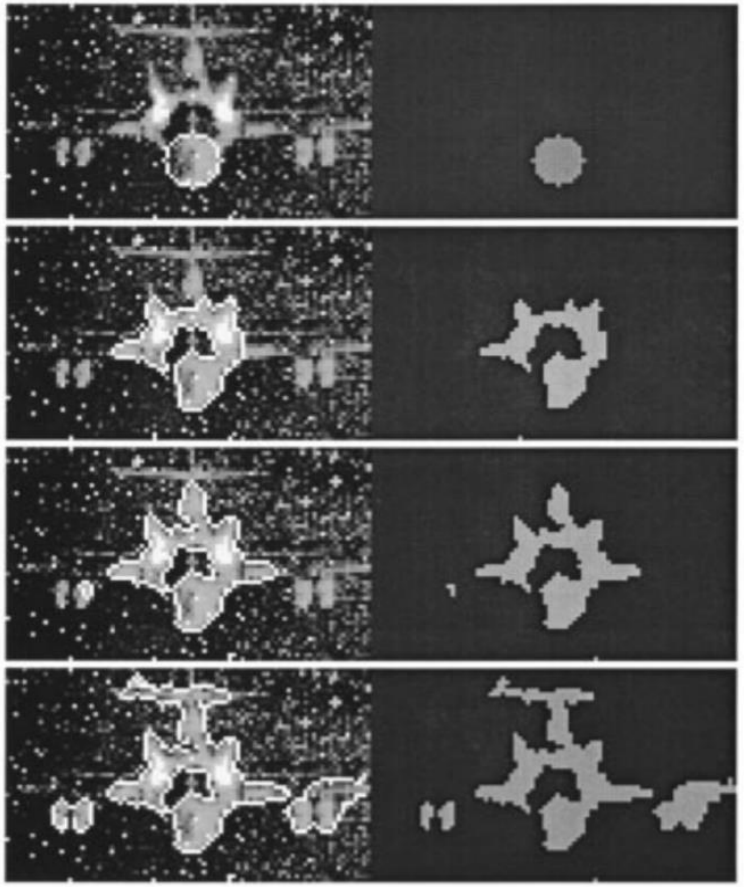
Minimize the energy functional

$$E(C) = \mu \cdot \text{length}(C) + \nu \cdot \text{area}(C)$$

$$+ \lambda_1 \int_{\text{Inside}(C)} |u_0 - c_1|^2 \, dxdy$$

$$+ \lambda_2 \int_{\text{Outside}(C)} |u_0 - c_2|^2 \, dxdy$$

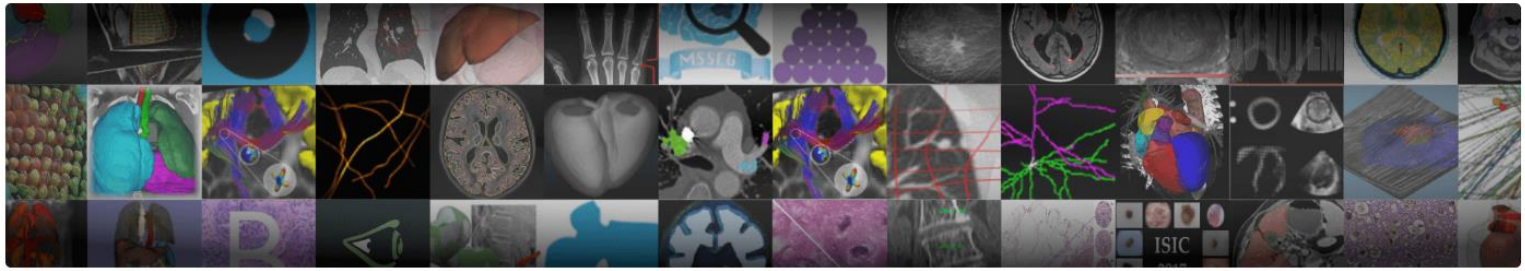
https://doi.org/10.1109/83.902291



How to Evaluate Segmentation Methods

Segmentation Method Evaluation

Grand Challenge All Challenges https://grand-challenge.org/ Sign in / Register



Grand Challenges in Biomedical Image Analysis

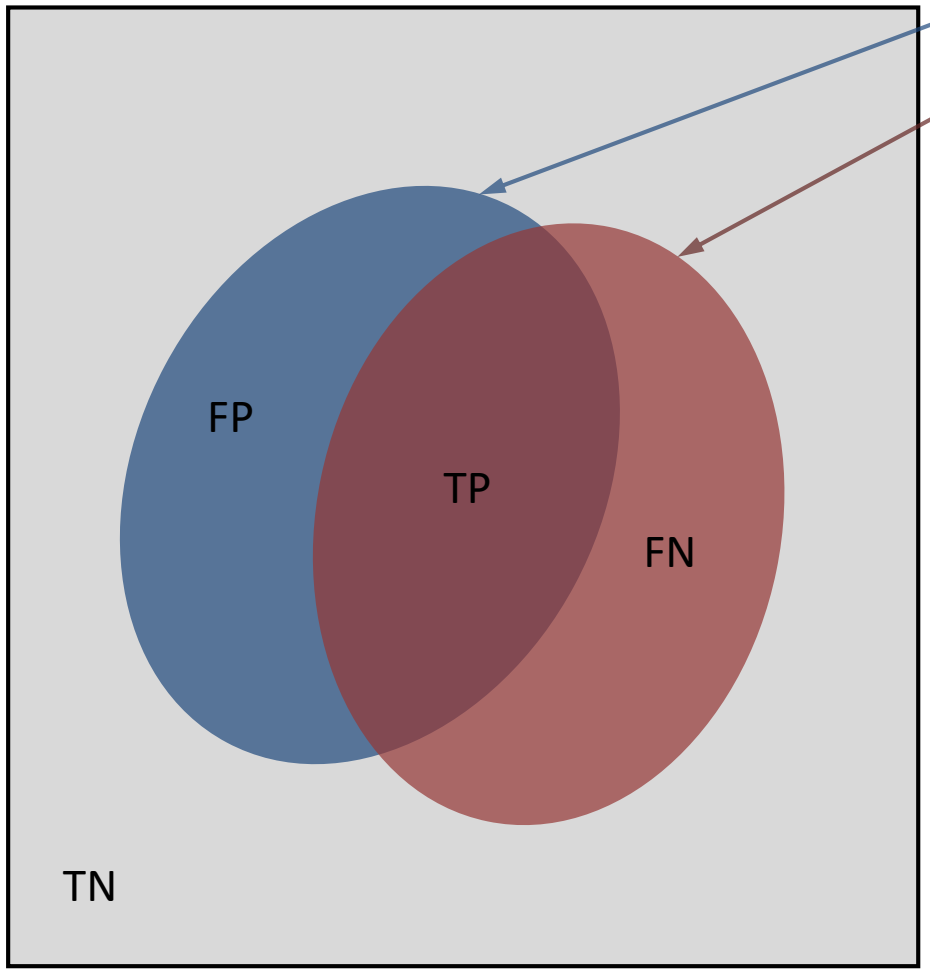
Every year, thousands of papers are published that describe new algorithms to be applied to medical and biomedical images, and various new products appear on the market based on such algorithms. But few papers and products provide a fair and direct comparison of the newly proposed solution with the state-of-the-art. We believe that such comparisons can help the research community and industry to develop better algorithms. We support the organization of these comparative studies and the dissemination of their results.

Organizing and participating in challenges is not the only way to facilitate better comparisons between new and existing solutions. If it were easy to publish and share your data, and the code you used to evaluate your algorithm's performance on that data, and possibly the algorithm itself, others could directly compare their approach to yours, using the same test data and the same evaluation metrics. With this site we provide tools to make it as easy as possible for you to publish your data and your evaluation for any paper you've written.

Why Challenges? describes the rationale for organizing grand challenges, provides advice for those who want to organize such events, and discusses where we hope the field will move to next.

All Challenges provides an overview of all previous, ongoing and upcoming challenges in biomedical image analysis that we are aware of. Drop us a note if you want your event listed on this overview.

Evaluation Metrics



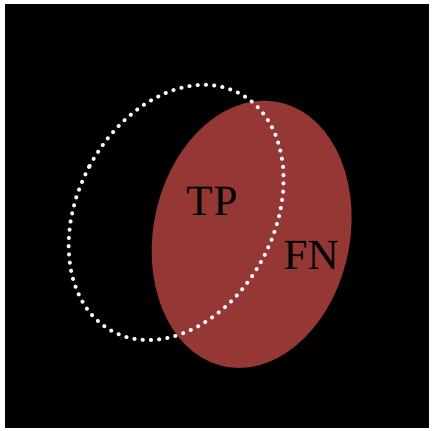
Segmented object pixels (set S)

True object pixels (set T)

Pixel classification

- True Positives $\mathrm{TP} = S \cap T$
  Correctly segmented as object
- True Negatives $\mathrm{TN} = S^c \cap T^c$
  Correctly segmented as background
- False Positives $\mathrm{FP} = S \cap T^c$
  Incorrectly segmented as object
- False Negatives $\mathrm{FN} = S^c \cap T$
  Incorrectly segmented as background

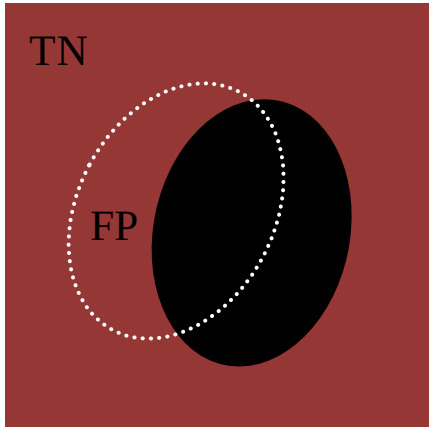
Evaluation Metrics



- Sensitivity (= true-positive rate)

  Fraction of the true object that is correctly segmented

$$TPR = \frac{|TP|}{|TP| + |FN|}$$



- Specificity (= true-negative rate)

  Fraction of the true background that is correctly segmented

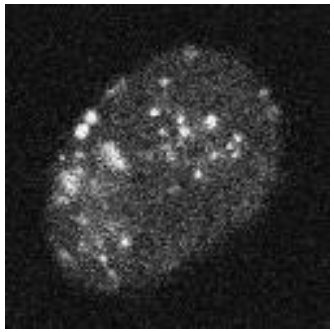
$$TNR = \frac{|TN|}{|TN| + |FP|}$$

Evaluation Metrics

- Receiver operating characteristic (ROC) of a method

  Plot of the true-positive rate (sensitivity) versus the false-positive rate (one minus the specificity) of a method as a function of its free parameter(s)

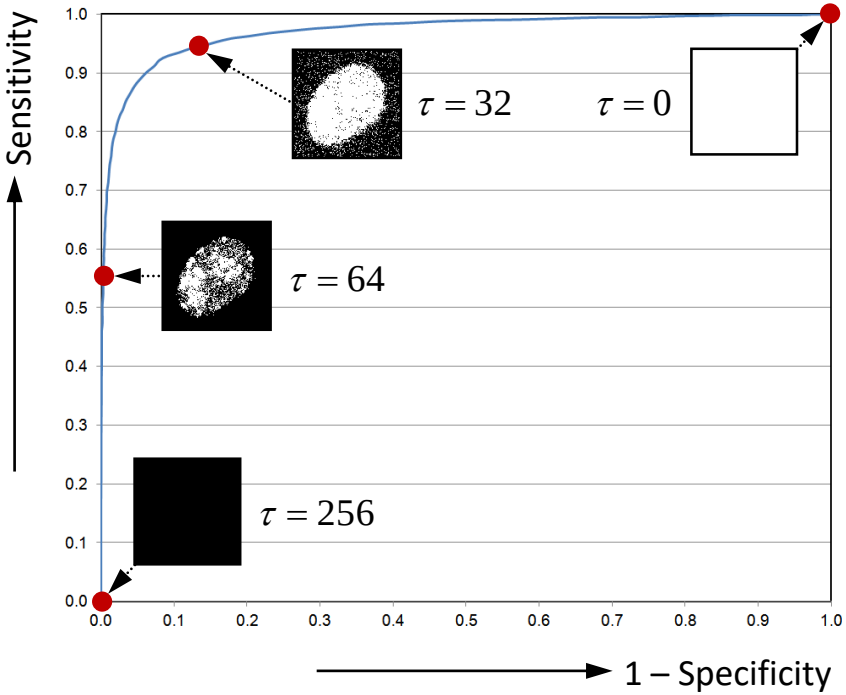
Example: Thresholding



Image

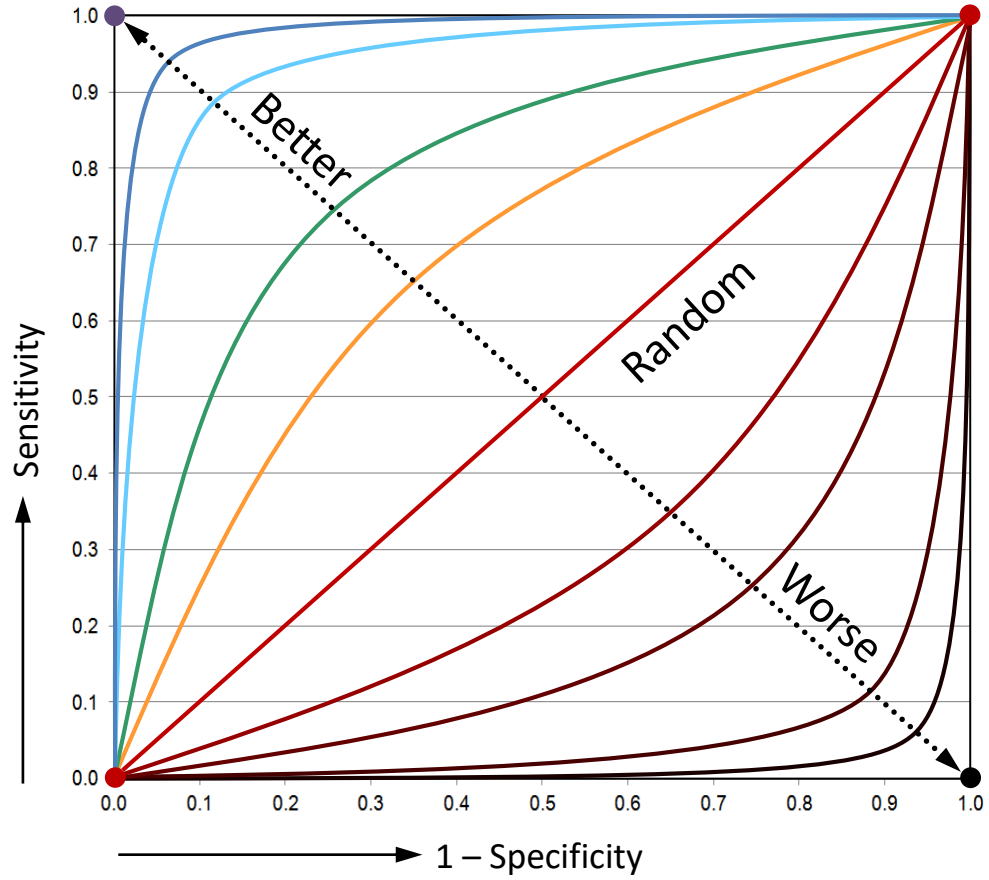
Truth

Compute the sensitivity versus the specificity for all possible intensity thresholds τ and plot the results



Evaluation Metrics

- Comparing the performance of methods by ROC analysis

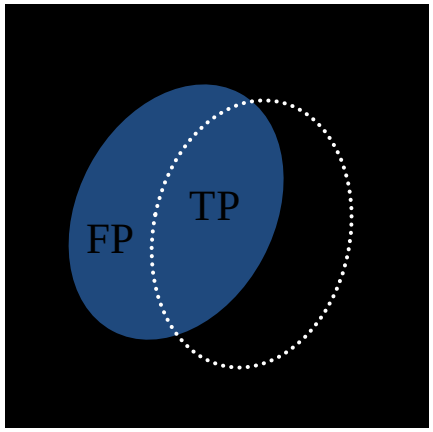


Area under the curve (AUC)

Worst	=	0.0
		0.1
		0.2
		0.3
		0.4
Random	=	0.5
		0.6
		0.7
		0.8
		0.9
Best	=	1.0

Higher AUC = better method

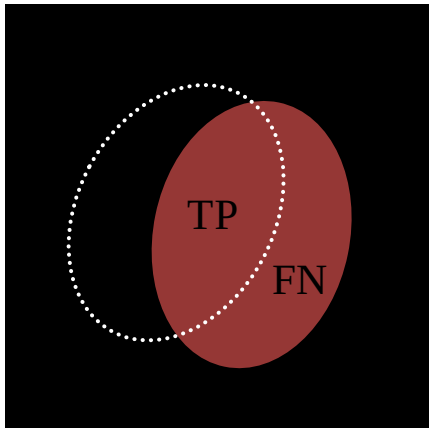
Evaluation Metrics



- Precision (= positive predictive value)

  Fraction of the segmented object that is correctly segmented

$$P = \frac{|TP|}{|TP| + |FP|}$$



- Recall (= sensitivity)

  Fraction of the true object that is correctly segmented

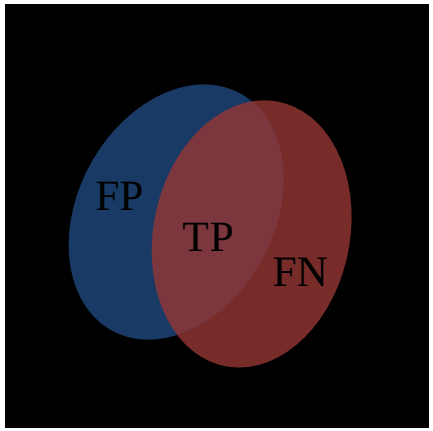
$$R = \frac{|TP|}{|TP| + |FN|}$$

- F-measure

  Harmonic mean of precision and recall

$$F = 2 \cdot \frac{R \cdot P}{R + P}$$

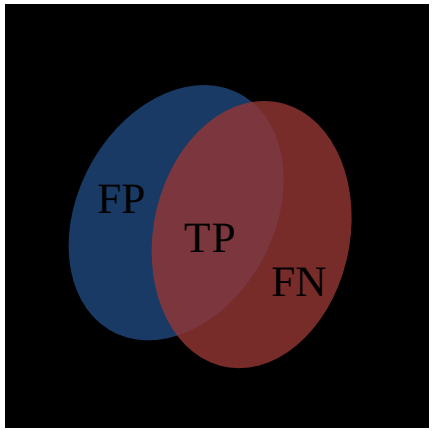
Evaluation Metrics



- Jaccard similarity coefficient (JSC)

  Fraction of the union of the segmented object and the true object that is correctly segmented

$$\mathrm{JSC} = \frac{|S \cap T|}{|S \cup T|} = \frac{|\mathrm{TP}|}{|\mathrm{FP}| + |\mathrm{TP}| + |\mathrm{FN}|}$$



- Dice similarity coefficient (DSC)

  Fraction of the segmented object set joined with the true object set that is correctly segmented

$$\mathrm{DSC} = 2\frac{|S \cap T|}{|S| + |T|} = \frac{2|\mathrm{TP}|}{|\mathrm{FP}| + 2|\mathrm{TP}| + |\mathrm{FN}|}$$

References and Acknowledgements

- Chapter 3 & 5 of Szeliski 2010
- Shapiro and Stockman 2001
- Some images drawn from Szeliski 2010
- Some images drawn from papers as indicated